Lic. Manuel Torres Remon

Desarrollo de aplicaciones web con PHP

- Lenguaje PHP
- Estructuras condicionales
- Estructuras repetitivas
- Funciones
- Arreglos

EDITORIAL MACRO

Sólo fines educativos - FreeLibros

Desarrollo de aplicaciones web con

PHP

EDITORIAL MACRO®

España - México - Colombia - Chile - Ecuador - Perú - Bolivia - Uruguay - Guatemala - Costa Rica

Sólo fines educativos - FreeLibros

Desarrollo de aplicaciones web con PHP
Autor: Manuel Ángel Torres Remon

© Derechos de autor registrados:
Empresa Editora Macro EIRL

© Derechos de edición, arte gráfico y diagramación reservados:
Empresa Editora Macro EIRL

Coordinación de edición:
Cynthia Arestegui Baca

Diseño de portada:
Alejandro Marcas León

Corrección de estilo:
Milton A. Gonzales M.

Diagramación:
Lizbeth R. Eufracio Quispe

Edición a cargo de:
© Empresa Editora Macro EIRL
Av. Paseo de la República N.° 5613 , Miraflores, Lima, Perú

Teléfono: (511) 748 0560
E-mail: proyecto@editorialmacro.com
Página web: www.editorialmacro.com

Primera edición: diciembre de 2014
Tiraje: 1000 ejemplares

Impresión
Talleres gráficos de la Empresa Editora Macro EIRL
Jr. San Agustín N.° 612-624, Surquillo, Lima, Perú

ISBN N.° 978-612-304-248-6
Hecho el depósito legal en la Biblioteca Nacional del Perú N.° 2014-16719

Prohibida la reproducción parcial o total, por cualquier medio o método, de este libro sin previa autorización de la Empresa Editora Macro EIRL.

Sólo fines educativos - FreeLibros

Manuel Ángel Torres Remon

El licenciado Manuel Ángel Torres Remon estudió en el Instituto de Educación Superior Manuel Arévalo Cáceres y en la Universidad Alas Peruanas. Actualmente se desempeña como consultor tecnológico de empresas como TopyTop y Nestlé. Asimismo, imparte los cursos de Programación, Análisis y Diseño de Sistemas y Base de Datos en instituciones superiores, como el instituto Manuel Arévalo Cáceres, Cibertec y Unimaster, de la Universidad Nacional de Ingeniería.

Ha publicado los libros de programación *Desarrollo de aplicaciones con Java*, *Programación orientada a objetos con Visual Basic 2012*, *Fundamentos de programación con Visual Basic 2012*, *Programación Transact con SQL Server 2012* y *Diseño web con HTML5 y CSS3*.

Para cualquier duda o sugerencia sobre el material, le puede escribir al siguiente correo electrónico: manuel.torresr@hotmail.com.

Sólo fines educativos - FreeLibros

Sólo fines educativos - FreeLibros

Agradecimientos

Cada libro desarrollado es una labor en equipo que a veces lo asume una sola persona, sacrificando su tiempo con la familia, horas de sueño y muchas otras cosas más, que luego tienen su recompensa cuando el material preparado es por fin una realidad.

Me complace compartir con ustedes los casos expuestos, los cuales además facilitarán mi desempeño como docente; por ello, en primer lugar, este agradecimiento, es para la familia MACRO, por confiar nuevamente en mi persona para la realización del presente libro.

En segundo lugar, deseo expresar todo mi agradecimiento a las personas que he privado de un tiempo que siempre les perteneció, como son mis pequeñas hijas Ángela Victoria y Fernanda Ximena Torres Lázaro y mi esposa Luz. Ellas, con mucha paciencia, me han brindado el tiempo necesario para la elaboración de este material. Siempre les estaré agradecido por ese gesto que han tenido conmigo.

Sólo fines educativos - FreeLibros

Dedicatoria

Este libro está dedicado a mis pequeñas Ángela y Fernanda, que son y seguirán siendo mi fuente de inspiración y mucho más. Y a mi esposa Luz por comprenderme en todo lo que me propongo.

Sólo fines educativos - FreeLibros

Índice

Introducción.....13

Capítulo 1

Introducción al HTML5.....15

1.1 Definiciones básicas.....17
1.1.1 *Software*.....17
1.1.2 *Software* libre.....18
1.1.3 Ventajas del *software* libre.....19
1.1.4 Desventajas del *software* libre.....19
1.1.5 *Software* propietario.....20
1.1.6 *Freeware*.....20
1.1.7 *Shareware*.....21
1.1.8 GNU.....21
1.1.9 Lenguaje interpretado.....21
1.1.10 Lenguaje compilado.....22
1.2 Introducción al HTML5.....23
1.2.1 Concepto.....23
1.2.2 Nuevos conceptos.....24
1.2.3 Estructura de una etiqueta HTML5.....25
1.2.4 Etiquetas obsoletas para HTML5.....26
1.2.5 Etiquetas HTML5 que cambian su significado.....26
1.2.6 Atributos de una etiqueta HTML5.....27
1.2.7 Identificación de los atributos en una etiqueta HTML5.....28
1.2.8 Especificación DOCTYPE.....29
1.2.9 La etiqueta HEAD.....29
1.2.10 La etiqueta BODY.....30
1.2.11 ¿Qué elementos podemos colocar dentro del BODY?.....30
1.2.12 Comentarios en HTML5.....32
1.3 Funcionamiento de un servidor web.....32
1.4 Introducción al Apache.....32
1.4.1 Descargar servidor Apache.....33
1.4.2 Instalación del servidor Apache.....34
1.4.3 Pruebas del servidor Apache.....37
1.5 Casos desarrollados de script HTML5 ejecutados desde el servidor Apache.....37
- Caso desarrollado 1: Menú de opciones vertical simple.....37
- Caso desarrollado 2: Menú de opciones horizontal.....40
- Caso desarrollado 3: Menú de opciones vertical con resaltado desde el puntero del mouse.....41
- Caso desarrollado 4: Sección con HTML5.....43
- Caso desarrollado 5: Artículo con HTML5.....44
- Caso desarrollado 6: Pie de página con HTML5.....45
- Caso desarrollado 7: Compra de productos con tablas.....46
- Caso desarrollado 8: Formulario de registro de usuarios.....48
- Caso desarrollado 9: Carga de archivos.....52

Sólo fines educativos - FreeLibros

Capítulo 2

Introducción al PHP 55

2.1 Definición de PHP 57
2.2 Usos de PHP 58
2.3 Evolución de PHP 60
2.4 Novedades de la última versión de PHP 60
2.5 Introducción al WAMP Server 61
2.5.1 Descargar WAMP Server 62
2.5.2 Instalación del servidor WAMP Server 63
2.5.3 Pruebas del servidor WAMP 66
2.5.4 Anomalías en la prueba del servidor WAMP 66
2.6 Instalación de Netbeans para PHP 68
2.6.1 Paquete de aplicaciones JDK 68
2.6.2 IDE Netbeans 70
2.7 Cuestiones posteriores a la instalación del Netbeans 73
• Cuestión 1: Crear un proyecto en Netbeans 73
• Cuestión 2: Agregar un archivo HTML5 al proyecto 75
• Cuestión 3: Agregar la paleta con etiquetas HTML5 76
• Cuestión 4: Agregar y ejecutar un archivo PHP 76
• Cuestión 5: Agregar y ejecutar un archivo de página web PHP 77
• Cuestión 6: Agregar un archivo CSS al proyecto 78
• Cuestión 7: Modificar la fuente y tamaño del código mostrado en el editor de Netbeans 79
• Cuestión 8: Modificar las ubicaciones de los proyectos al ejecutarlos 80
• Cuestión 9: Modificar el navegador predeterminado 81

Capítulo 3

Lenguaje PHP 83

3.1 Integrar código PHP en HTML5 85
3.2 Salida de información con PHP 86
3.2.1 Función echo 86
3.2.2 Función printf 87
3.2.3 Comentarios PHP 88
3.3 Página estática versus página dinámica 88
3.4 Manejo de literales de programación 91
3.5 Manejo de operadores 91
3.5.1 Operadores aritméticos 92
3.5.2 Operadores de cadena de caracteres 92
3.5.3 Orden de prioridad de los operadores 92
3.6 Manejo de variables 93
3.7 Tipos de datos usados en PHP 94
3.8 Manejo de constantes 96
3.9 Casos desarrollados 96
• Caso desarrollado 1: Diferencia entre echo y printf 96

Sólo fines educativos - FreeLibros

• Caso desarrollado 2: Manejo de variables y operadores....99
• Caso desarrollado 3: Manejo de constantes....102

Capítulo 4

Estructuras condicionales 105
4.1 Definición de lógica booleana 107
4.2 Estructurar una condición lógica en PHP....109
4.2.1 Operadores de comparación....110
4.2.2 Operadores lógicos 110
4.2.3 Estructurar bloques de código 110
4.2.4 Control de errores....111
4.3 Estructura condicional If simple 112
4.4 Estructura condicional If doble....113
4.5 Estructura condicional If doblemente enlazada 115
4.6 Estructura condición switch 117
4.7 Casos desarrollados....121
• Caso desarrollado 1: Salario de empleados usando condicional simple....121
• Caso desarrollado 2: Obsequio a clientes usando condicional simple....126
• Caso desarrollado 3: Venta de productos usando condicional doble....132
• Caso desarrollado 4: Control de mensualidad usando condicional doblemente enlazada...137
• Caso desarrollado 5: Venta de entradas usando condicional múltiple con switch....144

Capítulo 5

Estructuras repetitivas 149
5.1 Operadores de conteos y acumulaciones....151
5.1.1 Operadores de incremento y decremento....151
5.1.2 Operadores complejos....151
5.2 Contadores 152
5.3 Acumuladores 153
5.4 Estructura while....154
5.4.1 Ciclo de repeticiones while con cero iteración 156
5.4.2 Ciclo de repeticiones while infinito....156
5.4.3 Uso de la instrucción break en la estructura while 157
5.4.4 Uso de la instrucción continue en la estructura while 157
5.4.5 Anidamiento de ciclos while 158
5.5 Estructura for 158
5.5.1 Analogías entre while y for 161
5.5.2 Uso de la instrucción break en la estructura for 161
5.5.3 Uso de la instrucción continue en la estructura for 162
5.5.4 Anidamiento de ciclos for 162
5.6 Estructura do… while....163
5.6.1 Analogías entre while, for y do…while. 164
5.7 Casos desarrollados....164
• Caso desarrollado 1: Venta de productos usando while....165
• Caso desarrollado 2: Pago de préstamo usando for....171

Sólo fines educativos - FreeLibros

Capítulo 6

Funciones 177

6.1 Funciones para variables 179
6.1.1 Función isset 179
6.1.2 Función unset 181
6.1.3 Función gettype 182
6.1.4 Función settype 183
6.1.5 Función empty 184
6.1.6 Función is_integer 186
6.1.7 Función is_double 188
6.1.8 Función is_string 190
6.1.9 Función var_dump 192
6.2 Funciones de cadena 192
6.2.1 Función strlen 193
6.2.2 Función strpos 195
6.2.3 Función strcmp 196
6.2.4 Función strstr 196
6.2.5 Función substr 197
6.2.6 Funciones ltrim, rtrim, chop y trim 198
6.2.7 Función str_replace 198
6.2.8 Funciones strtolower y strtoupper 199
6.2.9 Función preg_match 199
6.2.10 Función explode 203
6.2.11 Función strrev 205
6.2.12 Función str_repeat 205
6.2.13 Función str_pad 205
6.3 Funciones numéricas 206
6.3.1 Función abs 206
6.3.2 Función ceil 207
6.3.3 Función exp 207
6.3.4 Función floor 207
6.3.5 Función getrandmax 208
6.3.6 Función max 208
6.3.7 Función min 208
6.3.8 Función mt_rand 209
6.3.9 Función pi 209
6.3.10 Función pow 209
6.3.11 Función round 210
6.3.12 Función sqrt 210
6.4 Funciones de fecha y hora 211
6.4.1 Función date 211
6.4.2 Función time 212
6.4.3 Función checkdate 213
6.4.4 Función getdate 214
6.5 Funciones implementadas por el usuario 215
6.5.1 Definición y usos 215
6.5.2 Implementación de una función 216

Sólo fines educativos - FreeLibros

6.5.3 Llamando a una función....218
6.5.4 Implementación de una función con parámetros....218
6.5.5 Implementación de una función con parámetros y con valor por defecto....219
6.5.6 Implementación de una función sin valor de retorno....220
6.5.7 Implementación de una función con múltiples valores de retorno....220
6.5.8 Implementación de funciones anónimas (lambda en PHP)....221
6.6 Funciones include y require....223
6.6.1 Función include....223
6.6.2 Función require....228
6.7 Casos desarrollados....232
• Caso desarrollado 1: Funciones de cadena - Registro de empleado....232
• Caso desarrollado 2: Funciones numéricas - Promedio de notas....240
• Caso desarrollado 3: Funciones implementadas por el usuario - Venta de productos....246
• Caso desarrollado 4: Funciones anónimas - Pago de estudiantes....255

Capítulo 7

Arreglos....261
7.1 Introducción....263
7.1.1 Tipos de arreglos....264
7.2 Estructura repetitiva foreach....266
7.3 Administrar elementos de un arreglo....269
7.3.1 Insertar elementos....269
7.3.2 Insertar elementos numéricos mediante una función....271
7.3.3 Recorrer los elementos por índice....272
7.3.4 Recorrer por elementos asociativos....274
7.3.5 Modificar elementos....275
7.3.6 Extrayendo un rango de elementos con array_slice....277
7.3.7 Avanzar y retroceder por elementos....277
7.3.8 Eliminar elementos....282
7.4 Métodos de un arreglo....285
7.4.1 Ordenamiento de elementos....285
7.4.2 Convertir un arreglo en una lista de variables....289
7.4.3 Convertir cadena de caracteres en array....293
7.4.4 Eliminando elementos repetidos en un array....294
7.4.5 Invertir los elementos de un arreglo....295
7.5 Arreglos multidimensionales....296
7.6 Casos desarrollados....299
• Caso desarrollado 1: Arreglo indexado - Informe de notas....299
• Caso desarrollado 2: Arreglo asociativo - Informe de notas....306
• Caso desarrollado 3: Arreglo indexado - Manejo de imágenes....312
• Caso desarrollado 4: Arreglo indexado - Paginación de productos....315
• Caso desarrollado 5: Arreglo indexado - Paginación de imágenes....318
• Caso desarrollado 6: Manejo de la función include - Listado de productos....322
• Caso desarrollado 7: Manejo de la función require - Control de pago....326
• Caso desarrollado 8: Manejo de la función require - Control de facturas....329

Sólo fines educativos - FreeLibros

Capítulo 8

Archivos 335

8.1 Manejo de archivos 337
8.1.1 Función file_exist 337
8.1.2 Función fopen. 338
8.1.3 Función fclose 339
8.1.4 Función fwrite 340
8.1.5 Función fread 340
8.1.6 Función fgets. 341
8.1.7 Función fputs. 342
8.1.8 Función rewind 343
8.1.9 Función filectime 343
8.1.10 Función file 344
8.1.11 Función file_put_contents 345
8.2 Manejo de archivos y carpetas 346
8.2.1 Función scandir 346
8.2.2 Función unlink 347
8.2.3 Función rename 347
8.3 Casos desarrollados 348
• Caso desarrollado 1: Verificar la existencia de un archivo 348
• Caso desarrollado 2: Contador de visita básico 352
• Caso desarrollado 3: Contador de visitas de forma gráfica 354
• Caso desarrollado 4: Control de registro de clientes 357

Capítulo 9

Sesiones 365

9.1 Introducción 367
9.2 Definición de sesiones 368
9.3 Funciones de session 368
9.3.1 Función session_start() 368
9.3.2 Función session_id() 369
9.3.3 Función session_name() 370
9.3.4 Función session_unset() 370
9.3.5 Función session_destroy() 370
9.4 Escritura y lectura de una variable de session. 371
9.5 Escritura y lectura de un arreglo unidimensional en la session. 372
9.6 Escritura y lectura de un arreglo asociativo en la session. 373
9.7 Casos desarrollados 374
• Caso desarrollado 1: Verificación de la session 375
• Caso desarrollado 2: Uso de colores desde la session 376
• Caso desarrollado 3: Encuesta de inseguridad 379
• Caso desarrollado 4: Login de usuario 392
• Caso desarrollado 5: Votación de candidatas 399
• Caso desarrollado 6: Carrito de compras básico - Venta de productos 405
• Caso desarrollado 7: Registro de nuevos productos 416

Bibliografía 423

Sólo fines educativos - FreeLibros

Introducción

PHP es uno de los lenguajes de programación web más usados en la actualidad, que combina su código con HTML5, como Visual o Java en sus aplicaciones web, implementando aplicaciones dinámicas de manera profesional. Entre sus características más relevantes, podemos afirmar que se trata de un lenguaje multiplataforma que puede ser ejecutado en cualquier tipo de dispositivo con suficiente capacidad para conectarse a cualquier base de datos; además, posee una buena fuente de documentación en su sitio web oficial. Asimismo, es considerado como un *software* libre, ya que puede ser usado en cualquier ámbito. Muchas compañías han optado por el uso de PHP por su versatilidad y su fácil uso, como Yahoo INC, Wikipedia.org, Facebook.com, Sourceforge.org, Flickr.com, Joomla Wordpress, Drupal, Moodle, etc.

El libro está dividido en nueve capítulos. En el capítulo uno se hace un reconocimiento de las etiquetas HTML5, ya que cuando PHP implementa sus aplicaciones se integra al HTML, y este a su vez al CSS3. Cabe mencionar que a partir del capítulo tres se presentan casos desarrollados que incluyen ambas tecnologías. En el capítulo dos se expone una introducción al lenguaje de programación PHP y se explica la instalación de las aplicaciones necesarias para poder desarrollar otras con PHP desde un servidor Apache. En el capítulo tres se analiza el lenguaje PHP, como la declaración de variables, constantes y, principalmente, la integración entre PHP y HTML5. Asimismo, se presentan casos desarrollados paso a paso. En el capítulo cuatro se reconocen todas las estructuras condicionales que maneja PHP, como el if simple, doble, doblemente enlazado; y condicionales múltiples, como el manejo del switch. En el capítulo cinco se utilizan las estructuras repetitivas de PHP, las cuales permiten crear aplicaciones de manera dinámica, proponiendo casos desarrollados que usen for y while. En el capítulo seis se hace referencia a las funciones en PHP, pues resulta importante implementar aplicaciones con programación modular usando las principales funciones que presenta PHP, como funciones de cadenas, fechas, etc. En el capítulo siete se trata el tema de los arreglos y la diferencia que presentan estos con los lenguajes tradicionales, integrando a la vez el tema de funciones. En el capítulo ocho se desarrolla el tema de archivos para implementar aplicaciones que manejen información permanente, para lo cual se manejará nuevas funciones en el manejo de los archivos desde PHP. Finalmente, en el capítulo nueve se desarrolla el tema de sesiones, que permitirá crear aplicaciones web que sean capaces de fusionar las funciones y arreglos en una misma aplicación.

Sólo fines educativos - FreeLibros

Sólo fines educativos - FreeLibros

CAP.

1

Introducción al HTML5

Sólo fines educativos - FreeLibros

Sólo fines educativos - FreeLibros

PHP es considerado como un lenguaje de programación de código abierto, es muy usado por los desarrolladores de páginas web, también llamados *webmaster*. Así como PHP, existen muchos lenguajes que permiten implementar aplicaciones web, como Visual Studio con su ASP, o Java con sus Java Servlets y JavaServer Pages (JSP). Desde aquí se desprende el término código abierto, sobre el cual también se expondrá en este capítulo, ya que resulta importante diferenciar los tipos de *software* que hay en el mercado, y cuál de ellos implementa PHP. El común denominador entre todos los lenguajes mencionados es la integración de su lenguaje con las etiquetas HTML; en nuestro caso, lo realizaremos usando la tecnología propuesta por HTML5, para ello mencionaremos en este libro las principales etiquetas y su trabajo embebido con PHP.

Una de las ventajas del desarrollo de aplicaciones con PHP es que resulta muy sencillo de entender su sintaxis, pues solo se requiere tener alguna experiencia en cualquier lenguaje de programación y un tiempo para leer la sintaxis propia de PHP; y todo ello se puede encontrar en Internet, tanto en español (http://www.php.net/manual/es/) como en inglés (http://www.php.net/manual/en/).

También debemos considerar la importancia que tiene el lenguaje HTML5 en la elaboración de aplicaciones web con PHP, pues resulta de vital importancia su integración, por tal motivo hemos preparado en este capítulo un resumen de las principales etiquetas HTML5 con casos desarrollados, lo cual nos servirá para realizar aplicaciones web con PHP de forma profesional.

Hoy en día se ha vuelto muy importante la interacción entre el usuario y las aplicaciones web; prácticamente se ha vuelto una necesidad el uso de las aplicaciones, pero cada vez con menos asistencia o capacitación; es decir, el usuario deberá usar su sentido común para poder interactuar de la mejor manera posible frente a las aplicaciones web, ya que estas no solo están en las computadoras personales sino también en dispositivos móviles, como celulares o tabletas; o más cerca aun, con los dispositivos de uso doméstico, como son los aparatos electrónicos, por ejemplo, la televisión *smart* o el juego de consola PlayStation. Es así como PHP se convierte en una herramienta importante para la implementación de aplicaciones web.

1.1 DEFINICIONES BÁSICAS

Antes de comenzar con el desarrollo de aplicaciones con PHP debemos tener en cuenta algunos conceptos básicos que serán de vital importancia para nuestro desempeño en el uso del lenguaje PHP.

1.1.1 *Software*

El concepto varía dependiendo del entorno donde nos encontremos, pero veamos la definición expuesta en IEEE Software Engineering Standard: Glossary of Software Engineering Terminology. IEEE Computer Society Press, 1993: «Es el conjunto de los programas de cómputo, procedimientos, reglas, documentación y datos asociados, que forman parte de las operaciones de un sistema de computación».

O también podríamos decir que el *software* es un: «Conjunto de programas, instrucciones y reglas informáticas que permiten ejecutar distintas tareas en una computadora», según la Real Academia Española (RAE).

Finalmente, podríamos decir que el concepto de *software* abarca a todas las aplicaciones informáticas que tienen un objetivo claro y específico, como lo es PHP, que es considerado como *software* libre. Este concepto lo veremos en el siguiente tema. Veamos algunas imágenes de los programas más conocidos:

Sólo fines educativos - FreeLibros

Fuente: <http://windows.microsoft.com/>

Fuente: <http://www.linux.org/>

Fuente: <http://www.apache.org/>

Fuente: <http://www.php.net/>

1.1.2 *Software* libre

Hoy en día es muy usado el término *software* libre y su concepto resulta algo predecible, pues el término «libre» indica que se le otorga a los usuarios toda la libertad sobre el *software*. Se podría decir que el *software* se puede adecuar a una determinada situación; si no cumple con esta premisa, entonces no es considerado *software* libre. Así, podríamos tomar el siguiente concepto de *software* libre como válido: «Es el *software* que respeta la libertad de los usuarios y la comunidad. En grandes líneas, significa que los usuarios tienen la libertad para ejecutar, copiar, distribuir, estudiar, modificar y mejorar el *software*»[1].

Eso quiere decir que para considerar a un *software* realmente libre, este deberá venir con su código fuente y que los usuarios podrán modificarlos de acuerdo a lo que convengan. Hay que tener en cuenta que existe una gran diferencia entre *software* libre y *software* gratuito. Este último hace referencia a un costo cero por el *software*; pero no se considera como *software* libre, pues el término «libertad» no necesariamente es sinónimo del término «precio». Veamos algunos ejemplos de *software* libre:

Navegadores web:

- **Mozilla Firefox**: Navegador web considerado como un fuerte competidor de Internet Explorer. Una de las características principales es la rapidez al mostrar información, su entorno bastante amigable y su costo totalmente gratuito.
- **Google Chrome**: Navegador web de rápido servicio y cada vez más popular entre usuarios de todo el mundo. No solo trabaja en computadoras personales, ahora está disponible en dispositivos móviles, como celulares, tabletas y televisores inteligentes, y su distribución es gratuita.
- **Opera**: Navegador web considerado como el más completo y rápido. Cuenta con dos versiones, una propietaria y otra gratuita.

Clientes para protocolos de transferencia de archivos (FTP):

- **FileZilla**: Permite hacer transferencia de archivos desde una máquina local a un servidor web, de tal manera que podríamos subir nuestros documentos web directamente al servidor, sin necesidad de hacer un envío uno a uno de los archivos.

[1] Concepto tomado de: https://www.gnu.org/philosophy/free-sw.es.html.

Sólo fines educativos - FreeLibros

Editores de páginas web:

- **Amaya**: Es el editor oficial del W3C llamado también consorcio internacional de la web. Este permite implementar documentos web como si estuviera trabajando en Dreamweaver.
- **Nvu**: Es un editor de documentos web de forma visual, muy poco usado en nuestro medio, pero su característica principal es que en su mayor parte está basado en el motor de Mozilla.

1.1.3 Ventajas del *software* libre

Veamos algunas ventajas que puede presentar un *software* definido como libre:

- No permitirá la adquisición de licencias ni las generará, ya que de lo contrario no sería considerado como *software* libre.
- Puede representar un beneficio tecnológico para la sociedad, ya que es de uso libre e ilimitado.
- Evita la distribución con beneficio, ya que siempre será considerado como *software* libre en todos sus aspectos.
- Cumple con todos los estándares establecidos para un *software* y se encuentra actualizado, ya que existen muchos colaboradores en el mundo que se encuentran dispuestos a colaborar con el proyecto.
- Se caracteriza por ser diverso y no centrarse en un tema particular.

Muchos piensan que el *software* libre se caracteriza por ser inseguro, inestable y con poca tendencia a crecer, algo así como un temor de llevar todos los procesos que vienen funcionando correctamente a *software* libre. Pero hoy en el Perú está ocurriendo un crecimiento en el uso de este tipo de *software*, que es impulsado generalmente por el estado peruano, tal es así que los organismos públicos ya han comenzado a usarlo; dentro de los cuales se encuentran: la Superintendencia Nacional de Aduanas y de Administración Tributaria-Sunat, Ministerio de Educación, Poder Judicial, Ministerio del Trabajo, Ministerio Público, entre otros que ya operan sus procesos de negocios mediante el uso de *software* libre.

1.1.4 Desventajas del *software* libre

Veamos algunas desventajas que presenta el uso de un *software* que es considerado como libre:

- Se tiene que contar con una persona capacitada en el tema de *software* libre y con conocimientos de programación, ya sea PHP, JAVA, PERL o PYTHON.
- El *software* libre no ofrece ningún tipo de garantía sobre el uso o administración del mismo, ya que no cuenta con una empresa que respalda el uso de dicho *software*.
- La interfaz gráfica del usuario (GUI) recién está tomando un aspecto atractivo para el usuario final, asumimos que seguirá mejorando en los años siguientes.

Sólo fines educativos - FreeLibros

1.1.5 *Software* propietario

También es conocido como *software* no libre, *software* privativo, *software* privado, *software* con propietario o simplemente *software* de propiedad. Una de las características principales del *software* propietario es que no se tiene el código fuente del mismo; por lo tanto, no se puede adaptar a las necesidades de una determinada organización. Un *software* propietario pertenece a una determinada empresa, es ella la que mantiene los derechos de uso, alteración o distribución. Es aquí donde se desprende el término «licencia», el cual garantiza una actualización, capacitación e incluso un soporte técnico en beneficio del usuario.

Fuente: <http://www.omicrono.com/2012/10/microsoft-office-2013-llegara-a-ios-y-android-en-marzo/>

Fuente: <http://www.unocero.com/2013/09/13/sale-visual-studio-2013-release-candidate/>

1.1.6 *Freeware*

Viene del inglés y significa «*software* gratis», el cual presenta características como la distribución en forma gratuita, ya sea por medio de copias o descargas desde algún servidor. Es totalmente libre de uso; es decir, no se necesita licencia alguna y finalmente el tiempo de uso del freeware es limitado. En otros términos, el usuario decidirá hasta cuándo debe o no usar dicho *software*.

Podríamos nombrar algunos ejemplos de freeware, como el reproductor multimedia Windows Media Player, el reproductor iTunes, el navegador Internet Explorer, el navegador Opera Browser, Ares, Atube catcher, etc.

Fuente: <http://diymediahome.org/outputting-wmp-videos-to-second-display/?lang=es>

Fuente: http://www.izalmusic.com/tienda

Sólo fines educativos - FreeLibros

1.1.7 *Shareware*

Significa «compartir por partes», en la cual un determinado *software* puede ser distribuido entre los usuarios de forma gratuita, estos a su vez evaluarán dicho producto en un tiempo determinado; claro está que dicho *software* contará con las limitaciones puestas por el propietario.

Muchos propietarios de *software* han tomado este término como la capacidad de distribuir *software* de prueba, en lo cual se denota las limitaciones del *software* hasta que el usuario decida comprar una licencia del producto. Podemos observar *software* de distribución shareware en los antivirus que nos ofrecen usar su producto por un número determinado de días, al cabo de estos se solicita la adquisición del *software* a través de mensajes enviados desde el mismo *software* instalado.

1.1.8 GNU

GNU (Unix-like) es un sistema operativo implementado por el proyecto GNU, que fue iniciado por Richard Stallman, un programador estadounidense que es conocido por implantar un marco legal para el movimiento del *software* libre en el mundo. Gracias a él, hoy en día podemos hacer uso del *software* libre sin problemas de licencias, pero con ciertas limitaciones.

GNU está formado en su totalidad por *software* libre, pero la diferencia con el sistema operativo Unix es la libertad de su uso, y su carencia de código de Unix.

El sistema GNU se lanzó bajo una licencia denominada «copyleft», que representa sin autoría; por lo tanto, puede ser distribuido de manera que cualquier usuario es capaz ejecutar, copiar, modificar o distribuir el sistema sin ningún problema. Dicha licencia está contenida en la Licencia General Pública de GNU, más conocida como GPL.

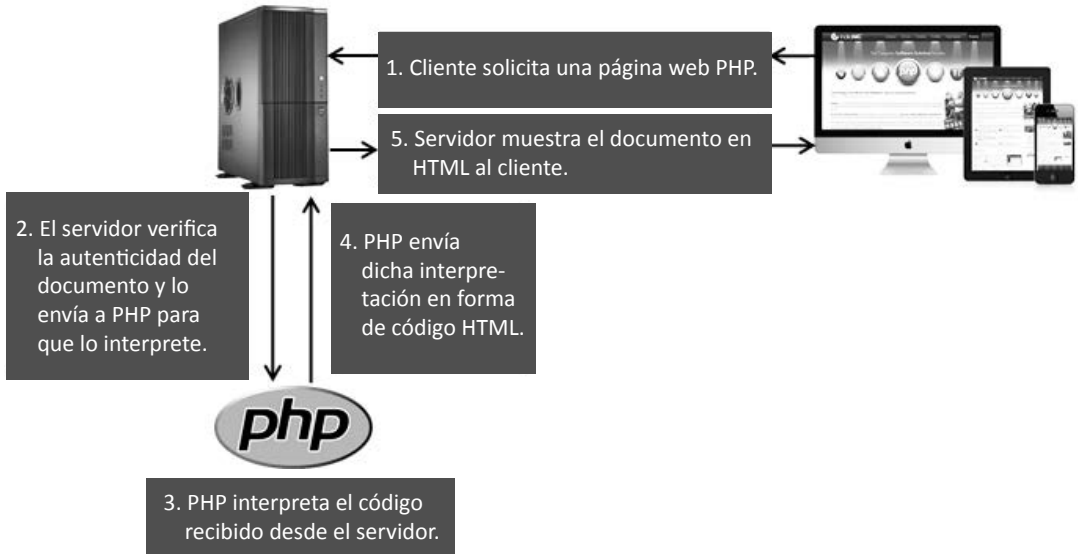
Richard Stallman además creó Free Software Foundation –también llamado «Fundación para el Software Libre» o simplemente FSF– para proveer soporte logístico, legal y financiero al proyecto GNU que tanto esperaba el mundo.

Es así que años más tarde, en 1991, un ingeniero de *software* llamado Linus Torvalds comenzó a implementar un núcleo que lo llamó Linux, y lo distribuyó bajo licencia GPL. Finalmente en 1992 es combinado con el sistema GNU, lo cual tuvo resultados nunca antes esperados; es así que se rompe el monopolio Microsoft y nace un pequeño monstruo llamado GNU/Linux, o simplemente el sistema operativo Linux.

1.1.9 Lenguaje interpretado

Se conoce como lenguaje interpretado cuando el código desarrollado es traducido por un intérprete a un lenguaje que puede ser entendido por la máquina, este proceso de interpretación se repetirá cada vez que se ejecute una determinada aplicación. Este lenguaje ha tenido un alto crecimiento en el desarrollo de aplicaciones web, como PHP, Ruby, Phyton y otros lenguajes interpretados.

Sólo fines educativos - FreeLibros



En la imagen observamos cómo se ejecuta una petición desde el cliente hacia el intérprete de PHP.

Veamos algunas características básicas:

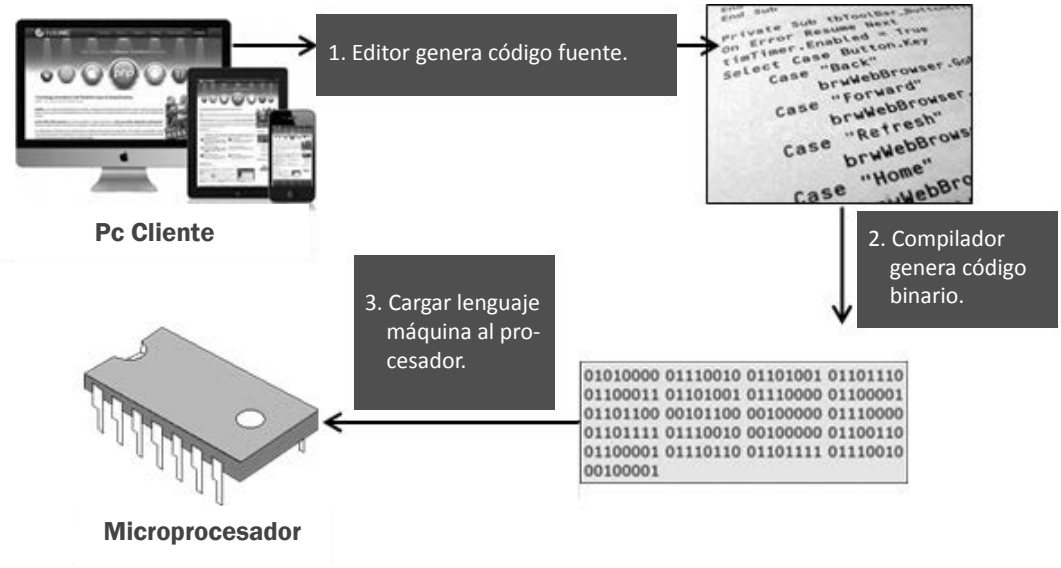
- Una variable puede almacenar un valor y comportarse como el tipo de datos que el valor le provee, es por eso que no es necesario realizar declaraciones de variables, ya que son declaradas como tipo de datos dinámicos.
- Posee independencia con respecto a la plataforma de trabajo; es decir, no tiene un *software* asociado al lenguaje; esto permite implementar dicho lenguaje en cualquier editor, como es el caso de PHP, el cual se puede codificar en Bloc de notas o Netbeans.
- La independencia de plataforma conviene al momento de comparar el tamaño de las aplicaciones, es por eso que los lenguajes interpretados mayormente son usados por aplicaciones web.
- Mayor consumo de tiempo porque necesita ser interpretado, en eso radica la diferencia con los lenguajes compilados que veremos más adelante.

1.1.10 Lenguaje compilado

Un lenguaje compilado hace que el código sea traducido por un compilador a un archivo ejecutable, entendible por una máquina desde cualquier plataforma de trabajo. Una de las grandes ventajas de este lenguaje compilado es que consume menos tiempo al ejecutar una aplicación, pues como el archivo ya se encuentra compilado ya no se ejecuta el proceso de interpretación a cada momento.

Entre los lenguajes de programación considerados como compilados, tenemos al lenguaje C con sus variaciones C++, Objective C, C#; y, por otro lado, Visual Basic. Este último es muy usado por programadores por ser un lenguaje compilado y muy sencillo de programar.

Sólo fines educativos - FreeLibros



En la imagen podemos observar cómo generar un código binario desde el compilador, que luego será cargado en el procesador de la computadora, para una mayor agilidad en el proceso de ejecución.

1.2 Introducción al HTML5

Fuente: <http://www.w3.org/html/logo/>

Fuente: <http://www.html5xcss3.com/2013/06/zcorporate-responsive-html5-theme.html>

1.2.1 Concepto

HTML5 es un lenguaje de etiquetas que permite diseñar documentos web estáticos; a diferencia de las versiones anteriores al HTML5, esta ofrece un conjunto de funciones que permitirá dar una nueva experiencia en el diseño web, veamos algunas de estas funciones:

- **Semántica**: Mejora en la distribución de los elementos web, lo que da precisión al contenido.
- **Conectividad**: Permite comunicarse con un servidor web de manera ágil, sin consumir muchas capacidades.
- **Servicio local**: Permite navegar por un sitio web sin necesidad de estar en línea, también es llamado ejecución local desde el lado cliente.
- **Multimedia**: Permite asignar archivos de videos o música con una sola etiqueta de manera sencilla.

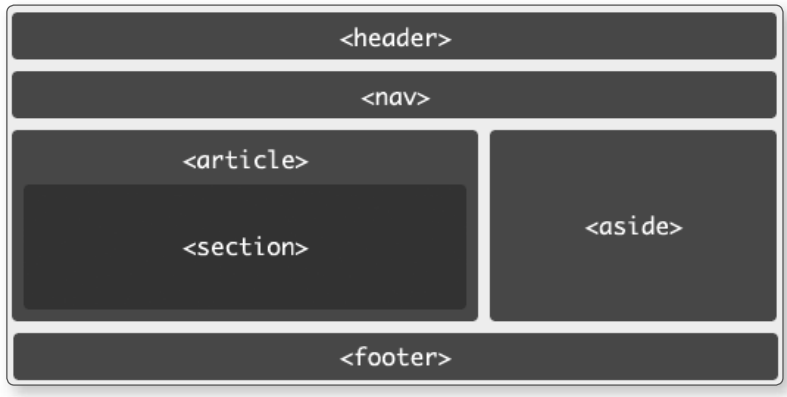
Sólo fines educativos - FreeLibros

- **Alto rendimiento**: Proporciona una mayor optimización con respecto a la velocidad usada en el *hardware* del equipo cliente.
- **Multiplataforma**: Permite visualizar las páginas web en diferentes dispositivos con la misma calidad.

1.2.2 Nuevos conceptos

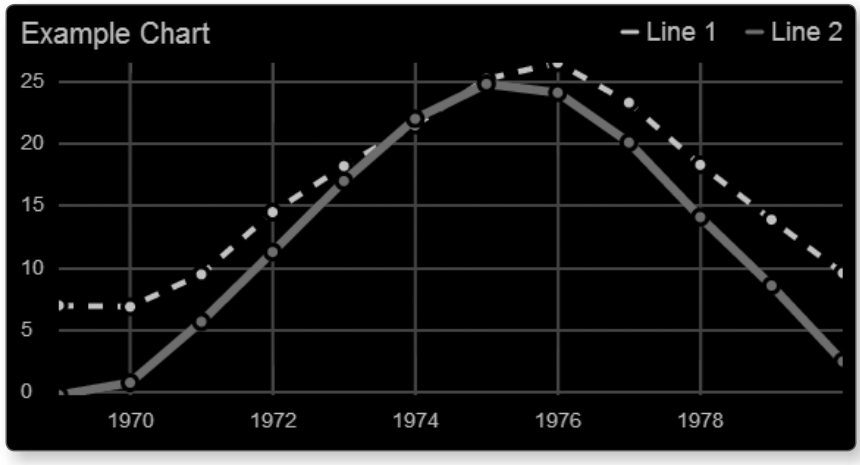
- **Con respecto a su estructura**:

  HTML5 presenta nuevas etiquetas que mejorarán la experiencia en el desarrollo de documentos web; eso quiere decir que las etiquetas anteriores al HTML5 han sido reformuladas en algunos casos, y otros se mantienen tal como se iniciaron. Esta nueva versión presenta un nuevo orden en los elementos; basado en seis etiquetas como su columna vertebral: header, nav, article, section, aside y footer, logrando a su vez un aumento en el dinamismo de la página web.



- **Gráficos canvas**:

  Canvas ya estaba implementado en las versiones anteriores al HTML5, pero esta vez mejora en el uso y propone nuevas funciones, siempre teniendo como base el script de Javascript.



Fuente: <http://www.html5canvastutorials.com/meteorcharts/html5-canvas-meteorcharts-series-styling/>

Sólo fines educativos - FreeLibros

- **Audio y video**:

  HTML5 incorpora un reproductor multimedia para la ejecución de videos y música en línea; a diferencia de las versiones anteriores al HTML5, esta funcionalidad presenta menos código para la implementación de audio y video en la web, haciéndolo sencillo y amigable. Además, no necesita la instalación de ningún *plugin* para su uso.

Fuente: <http://pixelcoblog.com/6-recursos-para-video-html5/>

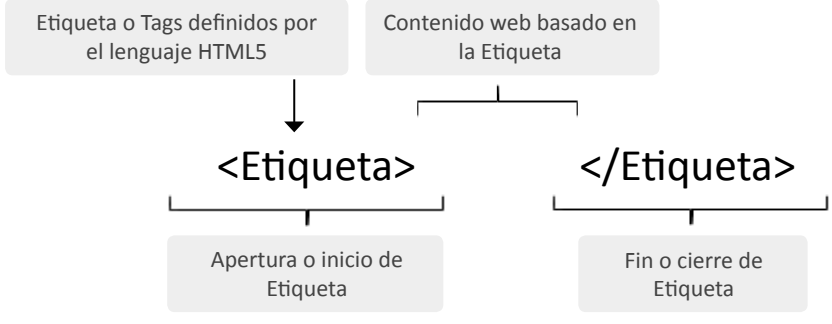
- **Geolocalización**:

  La geolocalización permitirá determinar desde qué lugar se está visualizando un sitio web, algo que ya se viene manejando en la red social Facebook. Y todo eso gracias a los sistemas de referenciación, como el GPS. Esto permitirá al usuario tener una rápida ubicación de la empresa que promociona la web, lo que dará confianza al cliente al momento de realizar una compra en línea.

Fuente: <http://www.cristalab.com/blog/como-funciona-la-geolocalizacion-por-wifi-c107677l/>

1.2.3 Estructura de una etiqueta HTML5

La estructura de las etiquetas HTML5 no ha sufrido cambios con respecto a las versiones anteriores, su formato es:



Sólo fines educativos - FreeLibros

Características de las etiquetas:

- Toda etiqueta en HTML5 se encuentra estandarizada por la W3C, que se considera como una comunidad internacional que desarrolla estándares que aseguran el crecimiento de la web.
- No es considerado *case sensitive*; es decir, no distingue la mayúscula de la minúscula, por lo tanto no genera error cuando se escribe de una u otra forma.
- Casi siempre se usan dos etiquetas; la primera indicará el inicio de la etiqueta, mientras que la otra especificará el final de la misma.
- Las etiquetas obsoletas provenientes de las versiones anteriores no han sido deshabilitadas por los navegadores web actuales; es así que podemos seguir usándolas, pero recuerde que HTML5 promueve tanto el uso de nuevas etiquetas como el de estilos CSS3.

1.2.4 Etiquetas obsoletas para HTML5

Veamos una lista de las principales etiquetas que ya no serán usadas por HTML5:

Etiqueta obsoleta	Descripción	Etiqueta en HTML5
applet	Permite insertar un script realizado por algún lenguaje de programación, como por ejemplo Java e integrarlo a html.	Object
basefont	Permite determinar el tamaño de fuente predeterminado.	Uso de CSS3
big	Permite mostrar un texto en tamaño grande.	Uso de CSS3
center	Permite centrar un elemento específico.	Uso de CSS3
dir	Permite insertar una lista de directorios.	UL
font	Permite establecer el estilo de fuente de un texto.	Uso de CSS3
frame	Permite insertar un marco web.	-
frameset	Permite insertar un grupo de frames o marcos web.	-
strike	Permite mostrar el texto que contiene estilo tachado.	Uso de CSS3
tt	Permite mostrar el texto como 'teletype' o espaciado simple.	Uso de CSS3
u	Permite mostrar el texto como subrayado.	Uso de CSS3
xmp	Permite definir un texto preformateado.	Pre

El uso de estilos CSS3 ha hecho que muchas etiquetas HTML ya no sean aplicables; es decir, perdieron su sentido de uso. El CSS3 presenta de manera profesional el mismo aspecto con ciertas ventajas; por ello, usted no tendrá problemas en adaptarse al cambio.

1.2.5 Etiquetas HTML5 que cambian su significado

En HTML5 se han reacomodado la funcionalidad de unas pocas etiquetas, la mayoría de ellas trabaja como en las versiones anteriores. Veamos la lista de las etiquetas que modifican su funcionalidad:

- **Etiqueta small**: En HTML5, la etiqueta **small** se usa para hacer referencia a condiciones legales, informaciones de *copyright*, etc.

Sólo fines educativos - FreeLibros

```
<small>
      Todos los derechos reservados-Copyright @2015.
</small>
```

- **Etiqueta b**: En HTML5 la etiqueta **b** se empleará cuando deseemos resaltar algún aspecto de un párrafo. La norma indica que la etiqueta **b** debe usarse como último recurso, y es mejor emplear encabezados (h1... h6), em, strong, mark o estilos CSS3 en función del tipo de información que se desea resaltar.

```
<p>
Solo rendirán examen aquellos alumnos que hayan entregado la boleta de pago en
<b>Counter de Secretaría Académica</b> en las fechas indicadas.
</p>
```

- **Etiqueta i**: En HTML5 la etiqueta **i** permite aplicar el estilo cursiva a un texto. Se recomienda su uso para resaltar nombres propios, una frase en otro idioma, una idea o un pensamiento, en otros casos usar estilos CSS3.

```
<p>
Solo rendirán examen aquellos alumnos que hayan entregado la boleta de pago en
<i>Counter de Secretaría Académica</i> en las fechas indicadas.
</p>
```

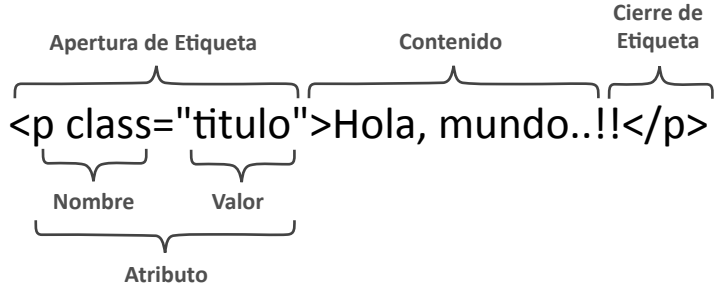
- **Etiqueta s**: En HTML5 la etiqueta **s** aplica un estilo tachado a un determinado texto el cual podría indicar que un párrafo no es recomendado o ya quedó obsoleto.

```
<p>Visítenos</p>
<p><s>dirección anterior: Av. Lima 549</s></p>
<p><strong>Nueva dirección: Jr. Cuzco 120</strong></p>
```

1.2.6 Atributos de una etiqueta HTML5

Se le denomina «atributos» a todas las opciones adicionales que puede tener una etiqueta; por ejemplo asignar un estilo o definir un nombre a la etiqueta, etc.

Componente de un atributo:



Sólo fines educativos - FreeLibros

1.2.7 Identificación de los atributos en una etiqueta HTML5

Veamos la descripción de los atributos pertenecientes a las etiquetas HTML5.

- **Atributo id**

  Este atributo permite asignar un identificador único a un elemento contenido en el script de una página web, lo cual es analizado por el Standard Generalized Mark-up Language (SGML) es muy parecido a la asignación de nombres que se le da a los objetos en un lenguaje de programación. Por ejemplo:

```
<p id="email">mtorres@queue-system.com</p>
<p id="movil">(051) 982-948948</p>
```

- **Atributo class**

  El atributo **class** se puede usar como selector para hojas de estilo especialmente cuando se desea asignar información de estilo a un conjunto de elementos web. Por ejemplo:

  1.- Implementamos un estilo

```
<style>
    p.info        { color: green }
    p.advertencia { color: blue }
    p.error       { color: red }
</style>
```

  2.- Aplicamos el estilo

```
<p class="info"> Mensaje de Información </p>
<p class="advertencia"> Mensaje de Advertencia </p>
<p class="error"> Mensaje de Error </p>
```

- **Atributo title**

  Permite establecer un mensaje a un determinado elemento web; es decir, mostrará alguna información de apoyo al usuario sobre un determinado elemento, claro está que solo será visible cuando el usuario posicione el puntero del mouse sobre dicho elemento. Por ejemplo:

```
<a href="mailto:comentarios@queue-system.com"
   title="Servicio de atención al cliente">
   Remítanos sus comentarios
</a>
```

Sólo fines educativos - FreeLibros

1.2.8 Especificación DOCTYPE

La declaración DOCTYPE le informa al navegador web con qué versión de HTML está escrita dicha página, esto lo realiza basándose en la especificación DTD, más conocida como Document Type Definition; esta define las reglas usadas por HTML5 en un documento web y será interpretado correctamente por el servidor web. Su formato es:

```
<!DOCTYPE html>
```

Las versiones anteriores al HTML5 contaban con el siguiente formato:

- **Doctype HTML 4.01 strict**

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01//EN"
"http://www.w3.org/TR/html4/strict.dtd">
```

- **Doctype HTML 4.01 transitional**

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">
```

1.2.9 La etiqueta HEAD

La etiqueta HEAD permite definir la cabecera del documento web y normalmente contiene información que no es visible para el usuario. HEAD pertenece a la estructura principal de un documento web; por lo tanto, debe ser colocado antes de la etiqueta <body> y dentro de <html>.

El formato completo de un documento web en HTML5 es como se muestra en el siguiente script:

```
<!DOCTYPE html>
<html>
      <head>
            Cabecera del documento
      </head>
      <body>
            Cuerpo del Documento.
      </body>
</html>
```

Lo que podemos asignar en el bloque HEAD puede ser:

```
<TITLE> Libro HTML5 </TITLE>
<META charset="utf-8" />
<LINK rel="stylesheet" href="estilo.css" />
```

Donde:

- **Title**: Define el título de la ventana al ejecutar el documento web.
- **Meta charset="utf-8"**: Define el uso de tildes y eñes en el documento web.

Sólo fines educativos - FreeLibros

- **Link**: Permite asociar un archivo al documento web actual, por ejemplo, para asociar un documento web a un archivo CSS, tendríamos el siguiente código:

```
<link href="estilo.css" rel="stylesheet" >
```

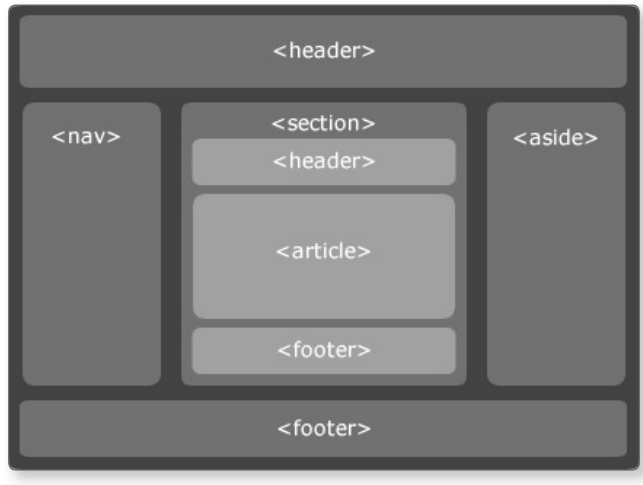
1.2.10 La etiqueta BODY

La etiqueta BODY es un espacio de trabajo que contiene todo lo visible en un documento HTML hacia el usuario; tal como textos, hipervínculos, imágenes, tablas, listas y todo lo que se pueda colocar en un documento web. La característica principal de la etiqueta BODY es que es compatible con todos los navegadores web actuales. La etiqueta BODY puede contener:

- Elementos estándares HTML5 como párrafos, formularios, imágenes, tablas, listas, etc.
- Enlaces a otras páginas web.
- Script de los lenguajes de programación como PHP, Java, ASP, etc.

1.2.11 ¿Qué elementos podemos colocar dentro del BODY?

El contenido de BODY ha sido mejorado para HTML5 haciéndolo semántico; es decir, permite programar en HTML5 con sentido semántico y entendible. Veamos las etiquetas relevantes de HTML5:



- **HEADER**

  Es la etiqueta que conforma la cabecera de una sección o artículo de un documento web basado en HTML5.

- **NAV**

  Esta etiqueta sirve para incluir menús de navegación en diferentes partes de nuestro sitio.

Sólo fines educativos - FreeLibros

- **SECTION**

  Permite crear secciones dentro de las demás etiquetas, organizando de la mejor manera la información que se muestre en un documento web.

- **ARTICLE**

  Un artículo es parecido a una sección en la cual podríamos colocar noticias, entradas del blog, trabajo de un portafolio o un contenido dinámico agregado con cierta frecuencia.

- **ASIDE**

  Es similar a una sección con la diferencia que se puede implementar como una barra lateral.

- **FOOTER**

  Es el pie de página de una sección, artículo, etc.

Veamos el script de una página web básica con HTML5:

```
<!DOCTYPE html>
<html lang="es">
<HEAD>
    <meta charset="utf-8" />
    <title>Estructura básica de una web - HTML5</title>
</HEAD>

<BODY>
     <HEADER>
        <NAV>
           <a href="opcion1.html"> Opcion 1 </a>
           <a href="opcion2.html"> Opcion 2 </a>
           <a href="opcion3.html"> Opcion 3 </a>
           <a href="opcion4.html"> Opcion 4 </a>
        </NAV>
      </HEADER>
      <ASIDE>
        <p> Contenido  aside </p>
      </ASIDE>
      <SECTION>
        <p> División de contenidos </p>
         <ARTICLE>
          <h1>Nombre del Articulo</h1>
          <p> Contenido del Articulo </p>
         </ARTICLE>
        </SECTION>
        <FOOTER>
            <p>Derechos reservados</p>
        </FOOTER>
</BODY>
</html>
```

Sólo fines educativos - FreeLibros

1.2.12 Comentarios en HTML5

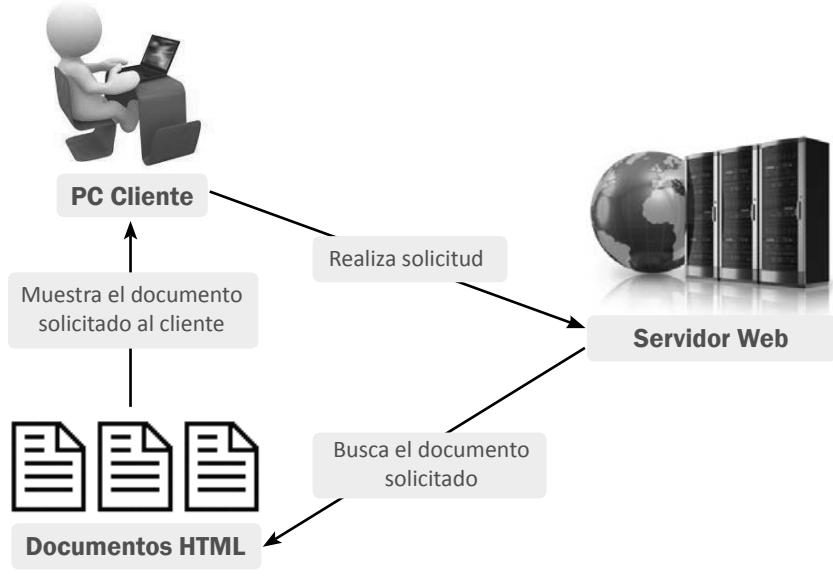
Un comentario permite asignar mensajes en un documento web con el fin de recordar algunos detalles del código, o simplemente mostrar un texto en cualquier parte del documento HTML5. El formato para asignar un comentario es:

```
<!-- Comentario de una línea-->
```

```
<!-- Comentario
     de
     varias
     líneas
-->
```

1.3 Funcionamiento de un servidor web

Un servidor web es aquel que presta servicios a los clientes, una de sus funciones principales es almacenar archivos pertenecientes a un sitio web y mostrarlo por la red, y así poder ser visitado por los usuarios en el mundo. Tal como se muestra en la siguiente imagen:



1.4 Introducción al Apache

El servidor Apache es considerado un servidor web de código abierto y de libre distribución; que puede ser usado en sistemas como Windows, Linux, Macintosh y otros.

Apache es una aplicación que permite implementar un servidor web en su computadora personal, asignándole un nivel de servidor local no importando el sistema operativo donde se encuentra, ya que tiene compatibilidad abierta. Su más cercana competencia es el Internet Information Server, más conocido como IIS, que pertenece a Microsoft y tiene las mismas funcionalidades de apache, pero es considerado un *software* propietario; por lo tanto solo funciona para sistemas Microsoft.

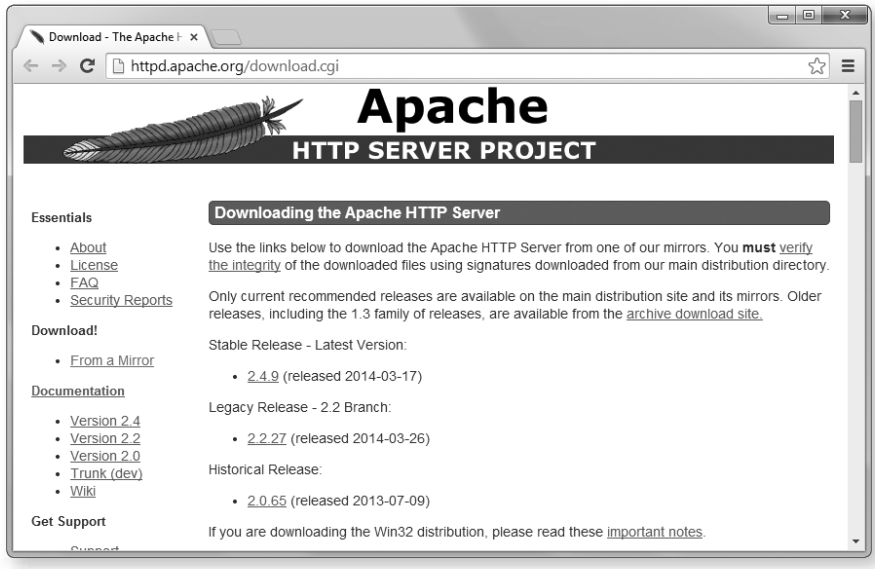
Sólo fines educativos - FreeLibros

Apache tiene una fuerte afinidad con el lenguaje de programación PHP por eso tiene librerías que soportan al PHP. Entre las características principales que presenta podemos mencionar:

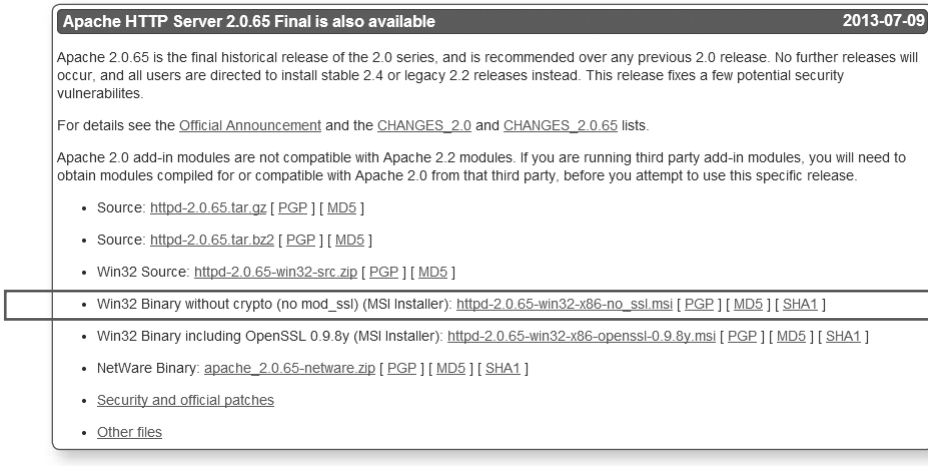
- Un completo soporte para el lenguaje de programación PHP.
- Incorpora módulos de autenticación web, como el **mod_access**, **mod_auth** y **mod_digest**.
- Presenta un soporte para certificados SSL y TLS.
- Permite la configuración de mensajes de errores personalizados y negociación de contenido.
- Permite autenticación de base de datos basada en SGBD.

1.4.1 Descargar servidor Apache

Para iniciar la descarga del servidor Apache, debemos ingresar a la siguiente URL **http://httpd.apache.org/download.cgi** tal como se muestra en la siguiente imagen:



Debemos considerar que el servidor Apache cuenta con licencia GPL; por lo tanto la descarga es *freeware*, y no estaríamos incurriendo en faltas.



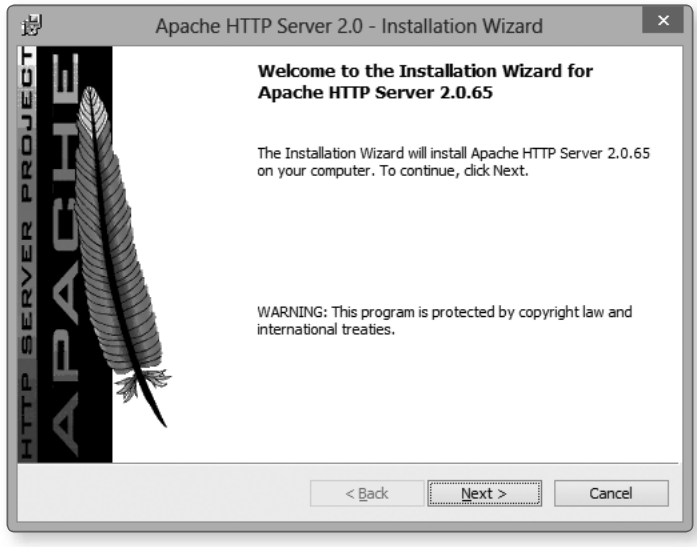
Sólo fines educativos - FreeLibros

En la página de descargas del servidor Apache debemos buscar la sección **Apache HTTP Server 2.0.65 Final is also available** y seleccionaremos httpd-2.0.65-win32-x86-no_ssl.msi, tal como se muestra a continuación:

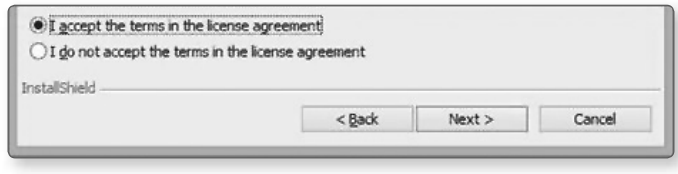
Finalmente la descarga se inicia, luego obtendremos el instalador del servidor Apache tal como se muestra en la siguiente imagen:

1.4.2 Instalación del servidor Apache

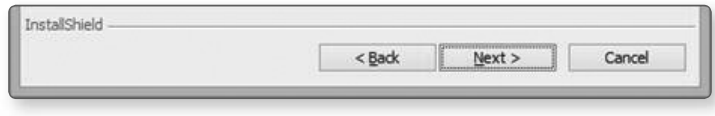
Vamos a iniciar el proceso de instalación del servidor Apache, la versión 2.0.65 ha sido probada para esta material usando el sistema operativo Windows 8.



Paso 1: Debemos seleccionar el botón **Next>** de la ventana de bienvenido al asistente de instalación de Apache.

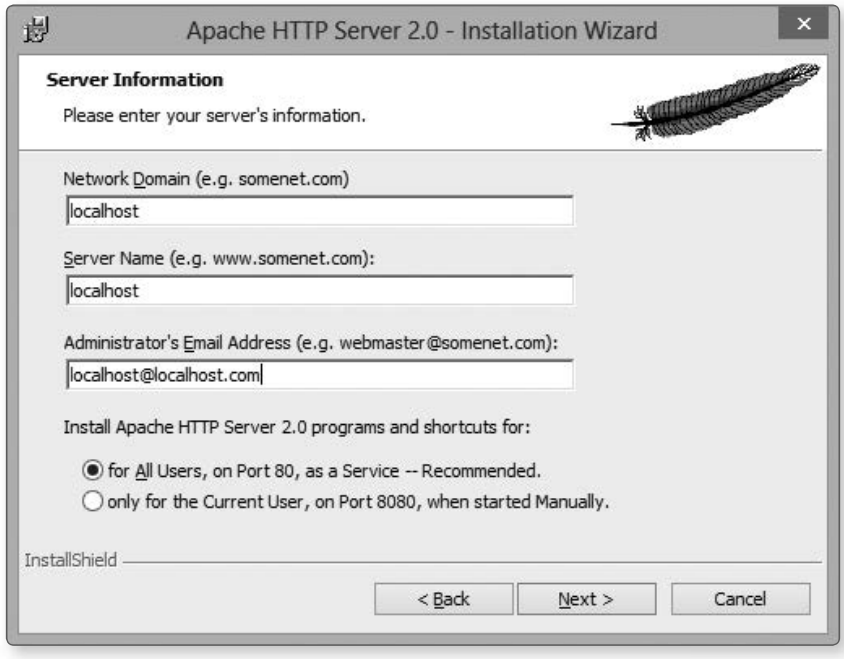


Paso 2: Seleccionar la opción «I accept the terms in the license agreement» y presionar el botón **Next>.**



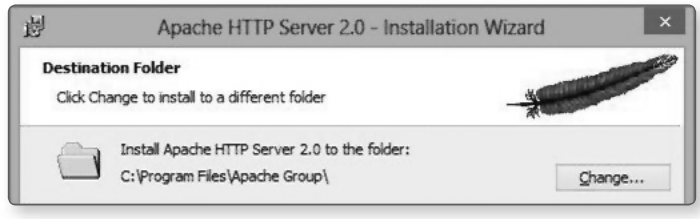
Sólo fines educativos - FreeLibros

Paso 3: En esta ventana solo presionar **Next>.**

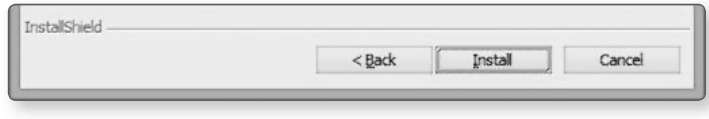


Paso 4: Debemos brindar información solicitada por el servidor de la siguiente manera: **Network Domain** colocar localhost, **Server Name** colocar localhost y **Administrator's Email Address** también asignar localhost@localhost.com.

Paso 5: Ahora nos toca seleccionar el tipo de instalación, para lo cual elegiremos **Typical**. No es necesario hacer una personalización en la instalación.

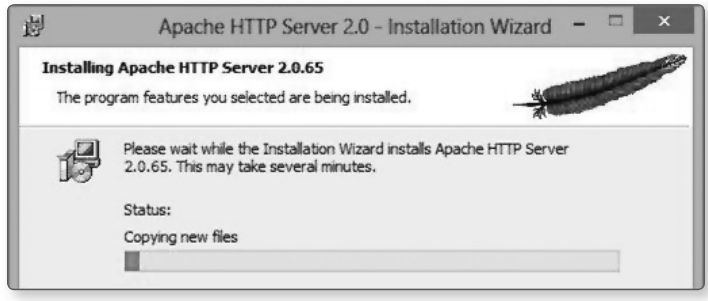


Paso 6: Seleccionaremos el lugar destino de la instalación, para ello notará que el asistente de instalación toma como carpeta predeterminada a **c:\Program files\Apache Group**; dejaremos dicha ruta como se muestra en la imagen, a menos que usted decida cambiarlo, para finalmente dar clic en el botón **Next>**.

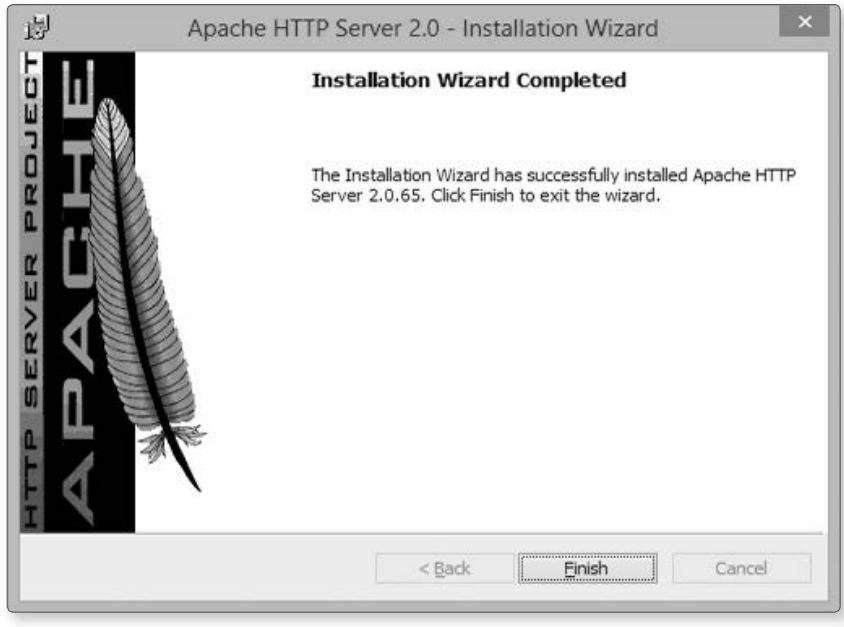


Sólo fines educativos - FreeLibros

Paso 7: Estamos listos para iniciar la instalación del servidor luego de haber seguido los pasos del asistente.



Paso 8: Vemos cómo se inicia el proceso de instalación del servidor Apache en el sistema operativo Windows 8.



Paso 9: Para finalizar la instalación debemos seleccionar el botón **Finish** de la ventana «Installation Wizard Completed».

Sólo fines educativos - FreeLibros

1.4.3 Pruebas del servidor Apache

Para comprobar si el servidor trabaja correctamente, debemos ingresar a un navegador web, como Internet Explorer, Mozilla Firefox o Google Chrome y colocar la siguiente **URL > localhost**, si todo es correcto usted deberá estar visualizando la siguiente imagen.

En caso no se muestre la imagen anterior, debemos revisar que otro servidor no se encuentre activo en nuestro sistema, como es el IIS (Internet Information Server) de Windows; y que para que funcione correctamente el Apache debemos inhabilitar el servicio IIS desde el panel de control de Windows.

1.5 CASOS DESARROLLADOS DE SCRIPT HTML5 EJECUTADOS DESDE EL SERVIDOR APACHE

Implementaremos casos prácticos de documentos web en HTML5 y CSS3 usando como servidor a Apache.

- **Caso desarrollado 1: Menú de opciones vertical simple**

Implemente una página web con HTML5 que permita mostrar un menú de opciones para el sistema de control de cliente, tal como se muestra en la siguiente imagen:

Sólo fines educativos - FreeLibros

Consideraciones:

- Las opciones del menú deben presentarse en una lista desordenada.
- Las opciones del menú deben presentarse de forma vertical.
- El enlace de cada menú será a un archivo con el mismo nombre, por ejemplo si la opción del menú es **Listado de clientes**, su enlace sería **listado.html**.

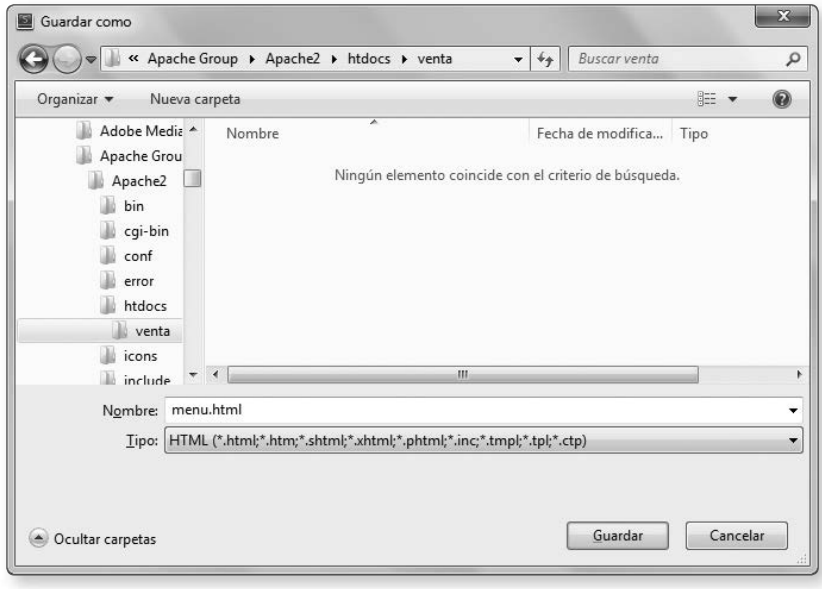
Solución:

Paso 1: Veamos el siguiente script implementado en la aplicación Sublime Text 2.0:

menu.html

```
<!DOCTYPE html>
<html lang="es">
  <head>
    <link rel="stylesheet" href="estilo.css" type="text/css">
    <meta charset="utf-8"/>
    <title>Sistema de Control de Clientes</title>
  </head>
  <body>
    <HEADER id="encabezado">
      <h1>Sistema de control de clientes</h1>
      <nav>
        <ul>
          <li><a href="principal.html">Principal</a></li>
          <li><a href="listado.html">Listado de clientes</a></li>
          <li><a href="registro.html">Registro del nuevo cliente</a></li>
          <li><a href="actualiza.html">Actualización de datos</a></li>
        </ul>
      </nav>
    </HEADER>
  </body>
</html>

```

Le asignaremos el nombre de **menu.html** y lo registraremos en la carpeta **htdocs** de Apache, el cual se encuentra ubicado en **Archivos de Programa > Apache Group > Apache2 > htdocs**. Es recomendado crear carpetas para una mejor administración de los documentos web, tal como se muestra en la siguiente imagen:

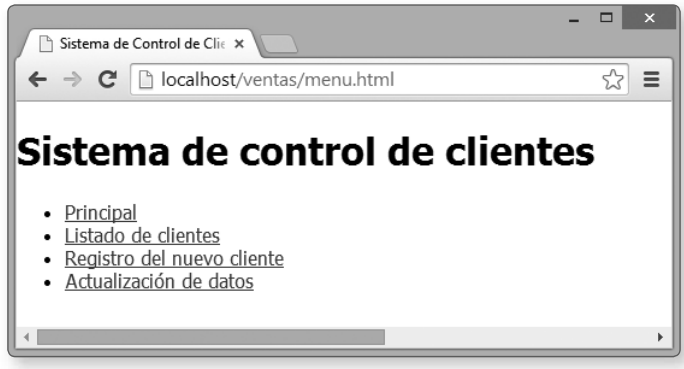


Sólo fines educativos - FreeLibros

Paso 2: Una vez colocado el archivo en la carpeta **htdocs**, procederemos a ejecutar el documento web, para lo cual debemos ingresar a un navegador web como podría ser Google Chrome, Mozilla Firefox o Internet Explorer y colocar el siguiente URL:

http://localhost/venta/menu.html

La siguiente imagen muestra el resultado de invocar al archivo **menu.html** ejecutado desde el servidor Apache.



Paso 3: De la misma manera podríamos crear un archivo llamado **estilo.css** que formateará los textos mostrados en el archivo **menu.html**; el script sería de la siguiente manera:

No olvide guardar el archivo **estilo.css** en la carpeta venta del Apache; es decir, ahora tendremos dos archivos, tal como se muestra en la siguiente imagen:

Sólo fines educativos - FreeLibros

Caso desarrollado 2: Menú de opciones horizontal

Implemente una página web con HTML5 que permita mostrar un menú de opciones para el Sistema de control de cliente, tal como se muestra en la siguiente imagen:

Consideraciones:

- Las opciones del menú deben presentarse en una lista desordenada.
- Las opciones del menú deben presentarse de forma horizontal.
- Las opciones del menú presentan un aspecto aplicado con estilo en la cual se configura color de fondo y color al texto.
- El enlace de cada menú será dirigido a un archivo con el mismo nombre, por ejemplo si la opción del menú es **Listado de clientes** su enlace sería **listado.html**.

Solución:

Paso 1: Veamos el siguiente script implementado en la aplicación Sublime Text 2.0:

menuh.html

```
<!DOCTYPE html>
<html>
    <head>
        <link type="text/css" href="estiloh.css" rel="stylesheet">
        <meta charset="utf-8"/>
        <title>Sistema de Control de Clientes</title>
    </head>
    <body>
        <header>
            <nav id="menu">
                <ul>
                    <li><a href="principal.html">Principal</a></li>
                    <li><a href="listado.html">Listado de clientes</a></li>
                    <li><a href="registro.html">Registro del nuevo cliente</a></li>
                    <li><a href="actualiza.html">Actualización de datos</a></li>
                </ul>
            </nav>
        </header>
    </body>
</html>
```

Le asignaremos el nombre de **menue.html** y lo registraremos en la carpeta venta de **htdocs**, el cual se encuentra ubicado en **Archivos de Programa > Apache Group > Apache2 > htdocs > venta**.

Paso 2: Ejecutar el archivo **menue.html** colocando el siguiente URL en el navegador:

http://localhost/venta/menuh.html

Sólo fines educativos - FreeLibros

Paso 3: No debemos olvidarnos de dar el formato adecuado para la presentación del menú horizontal usando estilos, para esto debemos crear un archivo CSS que contenga el siguiente código:

menuh.html | estiloh.css

```
#menu li {
    display: inline-block;
}

#menu a {
    display: block;
    float: left;
    margin-right: 0.12em;
    padding: 1em 1.5em;
    background: #178EAE;
    box-shadow: inset 0px -2px 0px 0px rgba(0,0,0,0.25);
    text-decoration: none;
    text-shadow: 1px 1px 0px #0C749C;
    text-transform: uppercase;
    font-size: .90em;
    color: #FFFFFF;
}

body {
    margin: 0em;
    padding: 0em;
    background: #1EB0D7;
    font-family: "Tahoma";
    font-size: 10pt;
    color: #525252;
}
```

Le llamaremos **estiloh.css** y lo guardaremos en el mismo lugar que el archivo **menuh.html**; es decir, en la carpeta **venta** del servidor Apache.

Caso desarrollado 3: Menú de opciones vertical con resaltado desde el puntero del mouse

Implemente una página web con HTML5 que permita mostrar un menú de opciones para el sistema de control de cliente, tal como se muestra en la siguiente imagen:

Consideraciones:

- Las opciones del menú deben presentarse en una lista desordenada.
- Las opciones del menú deben presentarse de forma vertical.
- Al pasar por las opciones del menú, cambiará de aspecto.
- Las opciones del menú presentan un aspecto aplicado con estilo, en el cual se configura el color de fondo y el color del texto.
- El enlace de cada menú será dirigido a un archivo con el mismo nombre, por ejemplo, si la opción del menú es **Listado de clientes** su enlace sería **listado.html**.

Sólo fines educativos - FreeLibros

Solución:

Paso 1: Veamos el siguiente script implementado en la aplicación Sublime Text 2.0:

menue.html

```
<!DOCTYPE html>
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<link rel="stylesheet" href="estiloe.css">
<meta charset="utf-8" />
<title>Sistema de Control de Clientes</title>
</head>
<body>
    <aside>
        <NAV class="aside-nav">
         <ul>
            <li><a href="principal.html">Principal</a></li>
            <li><a href="listado.html">Listado de clientes</a></li>
            <li><a href="registro.html">Registro del nuevo cliente</a></li>
            <li><a href="actualiza.html">Actualización de datos</a></li>
         </ul>
        </NAV>
    </aside>
</body>
</html>
```

Le asignaremos el nombre de **menue.html** y lo registraremos en la carpeta venta de **htdocs**, el cual se encuentra ubicado en **Archivos de Programa > Apache Group > Apache2 > htdocs > venta**.

Paso 2: Ejecutar el archivo **menue.html** colocando el siguiente URL en el navegador:

http://localhost/ventas/menue.html

Paso 3: No debemos olvidarnos de dar formato adecuado para la presentación del menú usando estilos, para esto debemos crear un archivo CSS que contenga el siguiente código:

menue.html | estiloe.css

```
/* Estilo de Fondo */
body {
    background: #d3d3d3;
    font-size: 100%;
    margin: 0;
    font-family:tahoma;
    color: #444;
}

/* Estilo para la lista */
.aside-nav ul {
  background: #e8e8e8;
  padding: 0;
  width: 25%;
}

/* Estilo para los elementos de la lista */
.aside-nav li a {
  display: block;
  padding: 1.25em;
  border-bottom: 1px solid #D2D2D2;
}

/* Estilo al posicionar el puntero encima del enlace*/
.aside-nav a:hover {
  color: #fff;
}
```

Le llamaremos **estiloe.css** y lo guardaremos en el mismo lugar que el archivo **menue.html**; es decir, en la carpeta **venta** del servidor Apache.

Sólo fines educativos - FreeLibros

Caso desarrollado 4: Sección con HTML5

Implemente una página web con HTML5 que permita mostrar un comentario sobre el sistema de ventas, tal como se muestra en la siguiente imagen:

Consideraciones:

- Debe usar la etiqueta **section** del HTML5.
- Los textos del título, subtítulo y contenido presentan estilos CSS.

Solución:

Paso 1: Veamos el siguiente script implementado en la aplicación Sublime Text 2.0:

seccion.html | estilo_seccion.css

```
<!DOCTYPE html>
<html>
<head>
<link rel="stylesheet" href="estilo_seccion.css">
<meta charset="utf-8" />
<title>Manejo de la etiqueta SECTION</title>
</head>
<body>
<SECTION>
    <header>
        <h1>Sistema de Ventas</h1>
    </header>
    <article>
        <h3>Control de Clientes</h3>
        <p>
            Nuestro sistema permite tener un control de los clientes mediante el
            registro de sus datos vía formulario web; considerándolo de esta manera
            como parte importante de nuestra empresa.
        </p>
    </article>
</SECTION>
</body>
</html>
```

Le asignaremos el nombre de **seccion.html** y lo registraremos en la carpeta **venta** de **htdocs,** el cual se encuentra ubicado en **Archivos de Programa > Apache Group > Apache2 > htdocs > venta**.

Paso 2: Ejecutar el archivo **seccion.html** colocando el siguiente URL en el navegador:

http://localhost/venta/seccion.html

Sólo fines educativos - FreeLibros

Paso 3: No debemos olvidarnos de dar formato adecuado para la presentación de la sección usando estilos, para esto debemos crear un archivo CSS que contenga el siguiente código:

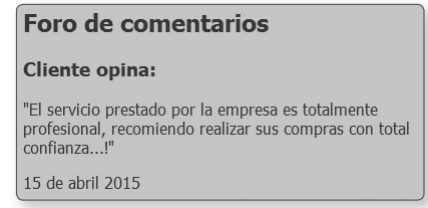
seccion.html | estilo_seccion.css

```
/* Estilo de Fondo */
body {
    background: #d3d3d3;
    font-size: 100%;
    margin: 10px;
    font-family:tahoma;
    color: #444;
}
/* Estilo para la seccion */
section{
    max-width:400px;
}
```

Le llamaremos **estilo_seccion.css** y lo guardaremos en el mismo lugar que el archivo **seccion.html**; es decir, en la carpeta **venta** del servidor Apache.

Caso desarrollado 5: Artículo con HTML5

Implemente una página web con HTML5 que permita mostrar un comentario realizado por un cliente sobre el sistema de ventas, tal como se muestra en la siguiente imagen:



Consideraciones:

- Debe usar la etiqueta **article** del HTML5.
- Debe usar la etiqueta **time** del HTML5 para mostrar la fecha actual.
- Los textos del título, subtítulo y contenido presentan estilos CSS.

Solución:

Paso 1: Veamos el siguiente script implementado en la aplicación Sublime Text 2.0:

articulo.html | estilo_articulo.css

```
<!DOCTYPE html>
<html>
<head>
    <link rel="stylesheet" href="estilo_articulo.css">
    <meta charset="utf-8" />
    <title>Manejo de la etiqueta ARTICLE</title>
</head>
<body>
 <ARTICLE>
  <header>
    <h1>Foro de comentarios</h1>
        <h3>Cliente opina:</h3>
  </header>
  <section>
        <p>
"El servicio prestado por la empresa es totalmente profesional, recomiendo realizar sus compras con total confianza...!"
      </p>
   </section>
   <p><time datetime="2015-04-15">15 de abril 2015</time></p>
 </ARTICLE>
</body>
</html>
```

Le asignaremos el nombre de **articulo.html** y lo registraremos en la carpeta **venta** de **htdocs**, el cual se encuentra ubicado en **Archivos de Programa > Apache Group > Apache2 > htdocs > venta**.

Sólo fines educativos - FreeLibros

Paso 2: Ejecutar el archivo **articulo.html** colocando el siguiente URL en el navegador:

http://localhost/venta/articulo.html

Paso 3: No debemos olvidarnos de dar formato adecuado para la presentación del artículo usando estilos, para esto debemos crear un archivo CSS que contenga el siguiente código:

articulo.html | estilo_articulo.css

```
/* Estilo de Fondo */
body {
    background: #d3d3d3;
    font-size: 100%;
    margin: 10px;
    font-family:tahoma;
    color: #444;
}
/* Estilo para el articulo */
article{
    max-width:400px;
}
```

Le llamaremos **estilo_articulo.css** y lo guardaremos en el mismo lugar que el archivo **articulo.html**; es decir, en la carpeta **ventas** del servidor Apache.

- **Caso desarrollado 6: Pie de página con HTML5**

Implemente una página web con HTML5 que permita mostrar el pie de página tal como se muestra en la siguiente imagen:

Consideraciones:

- Debe usar la etiqueta **footer** del HTML5.
- Debe controlar la alineación del texto en la misma línea, usando estilos CSS.
- Todos los textos presentan estilos CSS.

Solución:

Paso 1: Veamos el siguiente script implementado en la aplicación Sublime Text 2.0:

pie.html | estilo_pie.css

```
<!DOCTYPE html>
<head>
    <link href="estilo_pie.css" rel="stylesheet">
    <meta charset="utf-8" />
    <title>Manejo de FOOTER</title>
</head>
<body>
 <FOOTER>
    <p class="fl_left">Copyright &copy; 2015 –
                        Todos los derechos reservados -
      <a href="http://www.editorialmacro.com">Editorial Macro</a></p>
    <p class="fl_right">Diseñado por Lic. Manuel Torres R.</p>
 </FOOTER>
</body>
</html>
```

Le asignaremos el nombre de **pie.html** y lo registraremos en la carpeta **venta** de **htdocs**, el cual se encuentra ubicado en **Archivos de Programa > Apache Group > Apache2 > htdocs > venta**.

Sólo fines educativos - FreeLibros

Paso 2: Ejecutar el archivo **pie.html** colocando el siguiente URL en el navegador:

http://localhost/venta/pie.html

Paso 3: No debemos olvidarnos de dar el formato adecuado para la presentación del pie usando estilos, para esto debemos crear un archivo CSS que contenga el siguiente código:

pie.html | estilo_pie.css

```
/* Estilo para la alineacion de los textos*/
.fl_left{
    float:left;
}
.fl_right{
    float:right;
}

/* Estilo para todo el documento */
body{
    font-size:13px;
    font-family:Tahoma, Geneva, sans-serif;
    color:#FFF;
    background-color:#000;
}

/* Estilo para los enlaces */
a:link{ color:#FF0; }
a:visited{ color:#FF0;  }

/* Estilo para el pie */
footer {
    display:block;
    margin:10px;
}
```

Le llamaremos **estilo_pie.css** y lo guardaremos en el mismo lugar que el archivo **pie.html**; es decir, en la carpeta **venta** del servidor apache.

Caso desarrollado 7: Compra de productos con tablas

Implemente una página web con HTML5 que permita mostrar el resumen de compras realizada por un determinado cliente, tal como se muestra en la siguiente imagen:

Consideraciones:

- Debe usar tablas para la presentación de los productos.
- Los botones **actualizar** y **continuar** los puede obtener de la galería de imágenes del www.google.com.
- Todos los textos presentan estilos CSS.

Sólo fines educativos - FreeLibros

Solución:

Paso 1: Veamos el siguiente script implementado en la aplicación Sublime Text 2.0:

tabla.html

```
<!DOCTYPE html>
<html>
<head>
    <meta charset="utf-8"/>
    <title>Manejo de Tablas</title>
    <link href="estilo_tabla.css" rel="stylesheet"/>
</head>

<body>
<table width="700" border="0" cellspacing="0" cellpadding="0">
  <tr>
    <td width="24"> </td>
    <td width="655">
    <table width="700" BORDER="1" cellspacing="0"
           cellpadding="0" ID="CENTRADO">
      <tr ID="COLORFILA">
        <td>PRODUCTOS</td>
        <td>DESCRIPCION</td>
        <td>PRECIO</td>
        <td>CANTIDAD</td>
        <td>TOTAL</td>
      </tr>
      <tr>
        <td><img src="television.jpg" width="109" height="61"></td>
        <td>Television 42 pulgadas de alta resolución<br>
            COD: 2612215412</td>
        <td>S/. 1699.00</td>
        <td>01<br>
          <img src="actualizar.jpg" width="90" height="29"></td>
        <td>S/. 1699.00</td>
      </tr>
      <tr>
        <td><img src="blueray.jpg" width="108"
                 height="64"></td>
        <td>BlueRay SONY<br>
            COD: 2065150948198</td>
        <td>S/. 399.00</td>
        <td>01<br>
          <img src="actualizar.jpg" width="90" height="29"></td>
        <td>S/. 399.00</td>
      </tr>
      <tr  ID="COLORFILA">
        <td colspan="3"> </td>
        <td>SUBTOTAL</td>
        <td>S/. 2098.00</td>
      </tr>
    </table></td>
    <td width="21"> </td>
  </tr>
  <tr>
    <td> </td>
    <td><img src="continuar.jpg" width="128" height="39">
    </td>
    <td> </td>
  </tr>
</table>
</body>
</html>
```

Le asignaremos el nombre de **tabla.html** y lo registraremos en la carpeta **venta** de **htdocs**, el cual se encuentra ubicado en **Archivos de Programa > Apache Group > Apache2 > htdocs > venta**.

Paso 2: Ejecutar el archivo **tabla.html** colocando el siguiente URL en el navegador:

http://localhost/venta/tabla.html

Sólo fines educativos - FreeLibros

Paso 3: No debemos olvidarnos de dar el formato adecuado para la presentación de la tabla usando estilos, para esto debemos crear un archivo CSS que contenga el siguiente código:

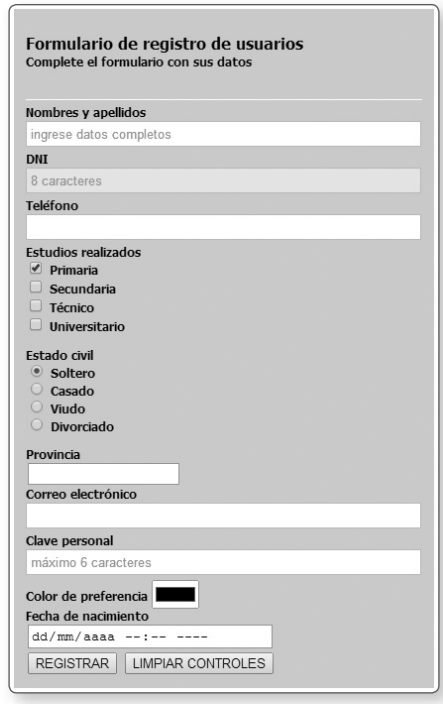
tabla.html | estilo_tabla.css

```
BODY{
    font-family:Tahoma, Geneva, sans-serif;
    font-size:12px;
}
#CENTRADO{
    text-align: center;
    color: #000;
}
#COLORFILA{
    background-color:#999;
}
```

Le llamaremos **estilo_tabla.css** y lo guardaremos en el mismo lugar que el archivo **tabla.html**; es decir, en la carpeta **venta** del servidor Apache.

Caso desarrollado 8: Formulario de registro de usuarios

Implemente una página web con HTML5 que permita mostrar un formulario de registro de usuario, tal como se muestra en la siguiente imagen:



Consideraciones:

- Debe usar la etiqueta **form** como plataforma del modelo.
- Todos los elementos son controles de formularios como caja de texto, botones, etc.
- Todos los textos y controles de formularios presentan estilos CSS.

Sólo fines educativos - FreeLibros

Solución:

Paso 1: Veamos el siguiente script implementado en la aplicación Sublime Text 2.0:

formulario.html

```
<!Doctype html>
<html>
<head>
    <title>formulario de registro de usuarios</title>
    <link href="estilo_formulario.css" rel="stylesheet">
    <meta charset="utf-8">
</head>
<body>
<section id="formulario">
 <form name="frmregistro" method="post" action="principal.html">
  <h1>Formulario de registro de usuarios</h1>
  <label id="label">Complete el formulario con sus datos</label>
      <p id="linea"> </p>

  <label id="label">Nombres y apellidos
     <input name="txtnombres" type="text" id="caja" size="60"
           placeholder="ingrese datos completos" required/>
  </label>
  <label id="label">DNI
     <input type="text" id="caja" name="txtdni" Size="60"
          placeholder="8 caracteres" disabled/>
  </label>
  <label id="label">Telefono
     <input type="tel" id="caja" size="60" name="txtfono"/>
  </label>
  <label id="label">Estudios realizados </label>
  <label id="label">
     <input type="checkbox" name="chkprimaria"
           checked="checked"/> Primaria
  </label>
  <label id="label">
     <input type="checkbox" name="chksecundaria" /> Secundaria
  </label>
  <label id="label">
     <input type="checkbox" name="chktecnico" /> Técnico
  </label>
  <label id="label">
     <input type="checkbox" name="chkuniversitario" /> Universitario
  </label>
  <br/>
  <label id="label">Estado civil</label>
  <label id="label">
      <input type="radio" name="btnestado" value="s"
            Checked="checked"/> Soltero
  </label>
  <label id="label">
      <input type="radio" name="btnestado" value="c" /> Casado
  </label>
  <label id="label">
      <input type="radio" name="btnestado" value="v" /> Viudo
  </label>
  <label id="label">
      <input type="radio" name="btnestado" value="d" /> Divorciado
  </label>
  <br/>

  <label id="label">Provincia</label>
      <input type="text" name="txtprovincia" list="provincias"/>
      <datalist id="provincias">
         <option value="Lima" />
         <option value="Arequipa" />
         <option value="Tacna" />
         <option value="Trujillo" />
         <option value="Huancayo" />
       </datalist>
```

Sólo fines educativos - FreeLibros

```
66    <label id="label">Correo electronico
67      <input type="email" id="caja" size="60" name="txtemail"/>
68        </label>
69
70        <label id="label">Clave personal
71      <input   type="password" id="caja"
72               name="txtclave" size="60"
73               Placeholder="máximo 6 caracteres "/>
74    </label>
75    <label id="label">Color de preferencia
76       <input type="color" name="txtcolor"/>
77    </label>
78    <label id="label">Fecha de nacimiento
79       <input type="datetime-local" name="txtfecha"/>
80    </label>
81    <input type="submit" value="REGISTRAR">
82    <input type="reset" value="LIMPIAR CONTROLES">
83   </form>
84  </section>
85  </body>
86  </html>
```

Le asignaremos el nombre de **formulario.html** y lo registraremos en la carpeta **venta** de **htdocs**, el cual se encuentra ubicado en **Archivos de Programa > Apache Group > Apache2 > htdocs > venta**.

Paso 2: Ejecutar el archivo **formulario.html** colocando el siguiente URL en el navegador:

http://localhost/venta/formulario.html

Paso 3: No debemos olvidarnos de dar el formato adecuado para la presentación del formulario usando estilos, para esto debemos crear un archivo CSS que contenga el siguiente código:

formulario.html | estilo_formulario.css

```
1  /*Estilo para el fondo de la pagina web*/
2  body{
3      font-family:"tahoma";
4      font-size:12px;
5  }
6
7  /*Estilo para el formulario*/
8  #formulario{
9      border:solid 2px #b7ddf2;
10     background:#b7ddf2;
11     margin:0 auto;
12     width:400px;
13     padding: 10px;
14 }
15
16 /*Estilo para el titulo*/
17 h1{
18     font-size:16px;
19     font-weight:bold;
20     margin-bottom:2px;
21 }
22
23 /*Estilo para el texto y su linea*/
24 #linea{
25     color:#666666;
26     margin-bottom:5px;
27     border-bottom:solid 1px #FFF;
28     padding-bottom:5px;
29 }
```

Sólo fines educativos - FreeLibros

formulario.html estilo_formulario.css

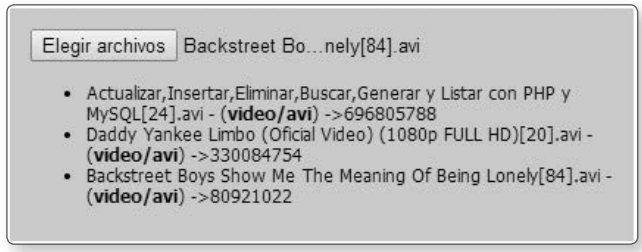
```
/*Estilo para el fondo de la pagina web*/
body{
    font-family:"tahoma";
    font-size:12px;
}

/*Estilo para el formulario*/
#formulario{
    border:solid 2px #b7ddf2;
    background:#b7ddf2;
    margin:0 auto;
    width:400px;
    padding: 10px;
}

/*Estilo para el titulo*/
h1{
    font-size:16px;
    font-weight:bold;
    margin-bottom:2px;
}

/*Estilo para el texto y su linea*/
#linea{
    color:#666666;
    margin-bottom:5px;
    border-bottom:solid 1px #FFF;
    padding-bottom:5px;
}
/*Estilo para los textos del formulario*/
#label{
    display: block;
    font-weight: bold;
    width: 240px;
    font-size: 12px;
}
/*Estilo para la cajas de texto simple*/
#caja{
    padding:4px;
    border:solid 1px #aacfe4;
    margin:2px 0 5px 0;
}

/*Estilo para el boton Registrar*/
#boton{
    clear:both;
    margin-left:150px;
    width:125px;
    height:31px;
    background:#666666;
    text-align:center;
    line-height:31px;
    color:#FFFFFF;
    font-size:11px;
    font-weight:bold;
}
```

Le llamaremos **estilo_formulario.css** y lo guardaremos en el mismo lugar que el archivo **formulario.html**; es decir, en la carpeta **venta** del servidor Apache.

Sólo fines educativos - FreeLibros

Caso desarrollado 9: Carga de archivos

Implemente una página web con HTML5 que permita mostrar los datos de los archivos que se cargan a un servidor web, tal como se muestra en la siguiente imagen:



Consideraciones:

- Use formulario para el registro de los archivos.
- Use código Javascript para obtener información de los archivos cargados en el servidor.
- Todos los textos presentan estilos CSS.

Solución:

Paso 1: Veamos el siguiente script implementado en la aplicación Sublime Text 2.0:

carga.html | estilo_carga.css

```
<!DOCTYPE html>
<html>
<head>
    <title>Registro de carga de archivos</title>
    <link href="estilo.css" rel="stylesheet">
    <meta charset="utf-8">
</head>
<body>
<section id="formulario">
    <input type="file" id="files" multiple>
    <ARTICLE id="archivos">
    <ul></ul>
    </ARTICLE>
<SECTION>
<script>
var insertar_en = document.querySelector("#archivos ul");
file_in = document.querySelector("#files")
file_in.onchange = function(e){
    var files = e.target.files;
    for(var i=0,f;f= files[i];++i){
        var archivo = document.createElement("li");
        archivo.innerHTML = f.name + " - (<b>" + f.type + "</b>) ->" + f.size;
        insertar_en.appendChild(archivo);
    }
}
</script>
</body>
</html>
```

Le asignaremos el nombre de **carga.html** y lo registraremos en la carpeta **ventas** de **htdocs**, el cual se encuentra ubicado en **Archivos de Programa > Apache Group > Apache2 > htdocs > venta**.

Sólo fines educativos - FreeLibros

Paso 2: Ejecutar el archivo **carga.html** colocando el siguiente URL en el navegador:

http://localhost/venta/carga.html

Paso 3: No debemos olvidarnos de dar el formato adecuado para la presentación de la carga de archivos usando estilos, para esto debemos crear un archivo CSS que contenga el siguiente código:

carga.html | estilo_carga.css

```
/*Estilo para el fondo de la pagina web*/
body{
    font-family:"tahoma";
    font-size:12px;
}

/*Estilo para el formulario*/
#formulario{
    border:solid 2px #b7ddf2;
    background:#b7ddf2;
    margin:0 auto;
    width:400px;
    padding: 10px;
}

/*Estilo para el titulo*/
h1{
    font-size:16px;
    font-weight:bold;
    margin-bottom:2px;
}
```

Le llamaremos **estilo_carga.css** y lo guardaremos en el mismo lugar que el archivo **carga.html**; es decir en la carpeta **venta** del servidor Apache.

Sólo fines educativos - FreeLibros

Sólo fines educativos - FreeLibros

CAP.

2

Introducción al PHP

Sólo fines educativos - FreeLibros

Sólo fines educativos - FreeLibros

PHP es el lenguaje de programación más usado en el mundo de la programación web, su desarrollo se basa mayormente en aplicaciones web; así como en registrar los datos de un usuario mediante un formulario, aplicar una encuesta a los usuarios sobre la preferencia por determinados productos, validar un usuario, etc.

También se dice que PHP convierte aplicaciones estáticas en dinámicas, como es el caso de aplicaciones realizadas puramente el HTML5, el cual solo es ejecutado en el lado cliente y no necesita intérpretes. PHP permite incluir su script en HTML5 para generar documentos dinámicos y crear aplicaciones robustas para la Web.

PHP fue creado por Rasmus Lerdorf en el año 1995 y desde su creación es considerado como *software* libre bajo la licencia de GNU. Es compatible con todos los sistemas operativos, incluyendo Microsoft Windows y Linux. Con ello nos sugieren que como programadores tenemos la libertad de escoger el sistema operativo en el que deseemos desarrollar las aplicaciones web con PHP.

Rasmus Lerdorf es un programador danés residente en Toronto, Canadá, y creó PHP a partir de la necesidad que sentía por saber cuántas personas visitaban su página web, en la cual exponía su hoja de vida. Esta pequeña aplicación ganó admiradores rápidamente pues era sencilla y fácil de entender, ya que era semejante a C o Java.

Fuente: <http://techwhizards.blogspot.com/2010/08/programming-language-inventors.html>

PHP incluye dentro de su lenguaje un «analizador sintáctico» que permite identificar las etiquetas HTML5, las interpreta y las remplaza por salidas esperadas por el usuario. Además incluye todas las funcionalidades que poseen los lenguajes de programación, como estructuras condicionales, repetitivas y funciones. Finalmente PHP incluye un interpretador de formularios web llamado inicialmente «Form Interpreter», soporte de nuevos protocolos de internet y, lo más importante, soporte de la mayoría de base de datos.

Inicialmente PHP era denominado como Personal Home Page Tools por el uso que le dio al inicio su creador, luego adquirió gran prestigio en el desarrollo de aplicaciones web, cambiando sus iniciales por Hypertext Pre-Processor; es así como se le conoce en la actualidad.

Gracias a la gran aceptación que ha tenido frente a desarrolladores de otros lenguajes, y a la colaboración de muchas personas mediante la comunidad, se ha logrado que PHP se convierta en un estándar en el mundo de la programación actual.

2.1 Definición de PHP

PHP viene de las palabras en inglés Hypertext Pre-Processor. Es considerado como un lenguaje de programación para aplicaciones web, se podría decir que su enfoque principal es desarrollar script que son interpretados por un servidor; es decir, es un lenguaje de programación interpretado.

- Logotipo oficial:

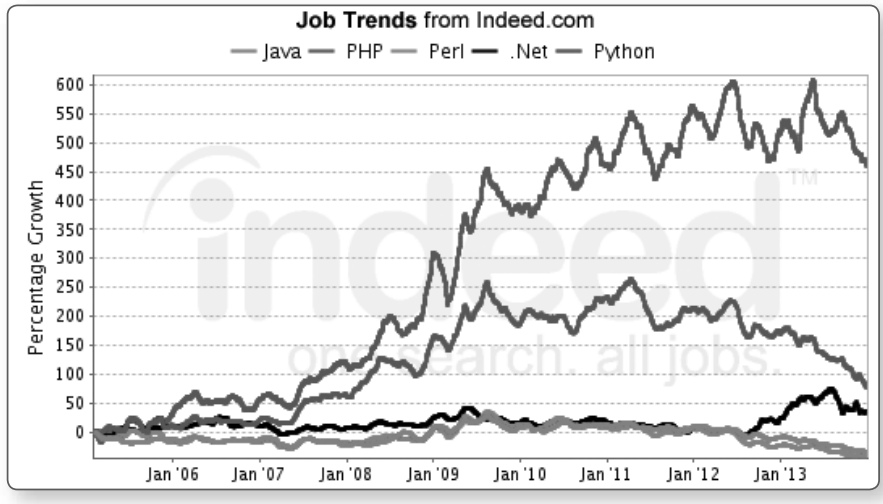
Sólo fines educativos - FreeLibros

- Última versión: 5.5.10
- Fecha de lanzamiento: Marzo 2014
- Requisitos:
  - Analizador de PHP (módulo CGI)
  - Servidor web
  - Navegador web

2.2 Usos de PHP

Explicaremos con gráficos el nivel de evolución y de empleabilidad de PHP frente a otros lenguajes de programación. El *ranking* de PHP frente a otros lenguajes de programación y, finalmente, en qué aplicaciones se ha usado PHP.

La empresa indeed.com muestra una estadística sobre la evolución de empleos en Estados Unidos entre los lenguajes de programación Java, PHP, Perl, .NET y Python. Tal como se puede ver, PHP es la segunda línea superior, según la información recogida hasta enero del 2013. Sabiendo que este material fue realizado en mayo del 2014, se sugiere que para ver los datos actualizados de la siguiente estadística visite la fuente de la siguiente imagen:



Fuente: <http://www.indeed.com/jobtrends?q=Java,+PHP,+Perl,+.Net,+Python&relative=1>

Sólo fines educativos - FreeLibros

La comunidad de programación TIOBE muestra el *ranking* de los lenguajes de programación más populares en el mundo, para esta evaluación considere que este material fue desarrollado en mayo del 2014, tal como se muestra en la siguiente imagen:

May 2014	May 2013	Change	Programming Language	Ratings	Change
1	1		C	16.926%	-1.80%
2	2		Java	16.907%	-0.01%
3	3		Objective-C	11.791%	+1.36%
4	4		C++	5.986%	-3.21%
5	7		(Visual) Basic	4.197%	-0.46%
6	5		C#	3.745%	-2.37%
7	6		PHP	3.386%	-2.40%
8	8		Python	3.057%	-1.26%
9	11		JavaScript	1.788%	+0.25%
10	9		Perl	1.470%	-0.81%
11	12		Visual Basic .NET	1.264%	+0.13%
12	10		Ruby	1.242%	-0.43%
13	38		F#	1.030%	+0.79%
14	14		Transact-SQL	1.025%	+0.21%

Fuente: <http://www.tiobe.com/index.php/content/paperinfo/tpci>

Las calificaciones se basan en el número de ingenieros cualificados en todo el mundo, cursos y proveedores de terceros. Los motores de búsqueda populares como Google, Bing, Yahoo!, Wikipedia, Amazon, YouTube y Baidu se utilizan para calcular las calificaciones. Hay que tener en cuenta que las aplicaciones más populares que hemos venido usando en estos últimos años están desarrolladas en PHP, o por lo menos una gran parte de ellas. Por ejemplo tenemos las siguientes aplicaciones:

- Wikipedia
- Facebook
- Taringa
- Joomla
- Wordpress
- YouTube
- Drupal
- Moodle

Sólo fines educativos - FreeLibros

2.3 Evolución de PHP

Veamos la evolución de PHP desde la versión 1.0, para lo cual mostraremos los respectivos años de lanzamiento de cada una de las versiones y algunas observaciones adicionales dependiendo de las mismas:

Versión	Año de lanzamiento	Observaciones
1	1995	Llamado Personal Home Page.
2	1997	Es considerado como una de las aplicaciones más rápidas y sencillas de desarrollar.
3	1998	El fundador de la empresa Zend Technologies Zeev Suraski reescribe parte del código PHP.
4.0	2000	Se incorpora al código PHP un «parsing» de dos fases llamado motor Zend.
4.1	2001	Se incorporan las variables globales $_GET, $_POST, $_SESSION.
4.2	2002	Se inhabilita register_globals en la instalación de PHP haciendo que el usuario lo habilite de forma manual.
4.3	Dic. 2002	Se incorpora una línea de comando llamada CLI.
4.4	Jul. 2005	Se incorporan páginas como phpsize y php-config.
5.0	2004	Se incorpora un nuevo modelo de objetos desde el motor Zend II.
5.1	Nov. 2005	Incorpora variables al compilador de PHP, haciendo una clara mejora en el rendimiento de sus aplicaciones.
5.2	2006	Incorpora un soporte especial para JavaScript Object Notation, más conocido como JSON.
5.3	2009	Incorpora un soporte para espacios de nombres y un enlace estático en tiempo de ejecución, funciones lambda y las funciones goto.
5.4	2012	Incorpora un soporte para JavaScript Object Notation, más conocido como Trait.
5.5	2013	Se incorporan nuevos generadores para bucles empty().
6.0	-	Esta versión está retrasada, ya que los creadores están corrigiendo las cadenas Unicode.

2.4 Novedades de la última versión de PHP

Nombraremos las novedades de la versión estable que presenta PHP hasta el día de la elaboración de este material; es decir, nombraremos las novedades de la versión 5.4.3 del PHP.

- **Namespace**: Permite tener control sobre nombres ambiguos en una determinada aplicación; es decir, clases, funciones o constantes no colisionarán al encontrar nombres similares.
- **Goto**: Es una función que permite realizar saltos dentro de una función o un procedimiento.
- **Lambda**: Es una función de tipo anónima pues no tiene un nombre de función. Esta función también es implementada en .NET y Java.
- **Trait**: Es un mecanismo que permite la reutilización de un determinado código que no tiene un soporte sobre la herencia múltiple, tal como sucede con PHP.

Sólo fines educativos - FreeLibros

2.5 INTRODUCCIÓN AL WAMP SERVER

Actualmente es llamado WAMP Server puesto que antes era simplemente WAMP, y es considerado uno de los paquetes más usados para la implementación de aplicaciones con PHP, pues incorpora un conjunto de aplicaciones como el servidor Apache, el servidor de base de datos MySQL y el lenguaje de programación PHP. De allí sus iniciales, la W hace referencia al sistema operativo Windows; eso quiere decir que también contamos con una versión para el sistema operativo Linux llamado LAMP.

Nombraremos algunas aplicaciones análogas al WAMP, estas se usan muchas veces dependiendo del sistema operativo o simplemente por elección del programador de aplicaciones:

XAMPP **Logotipo:** **Características:**	XAMPP • Cuenta con un servidor Apache. • Cuenta con el servidor MySQL. • Cuenta con los lenguajes de programación PHP y Perl. • Gestionar la transferencia de archivos FTP. • Gestiona el servidor de correos Mercury para envío de correos electrónicos. • Adicionalmente cuenta con el servidor Tomcat para aplicaciones servlets JSP. Su sitio oficial es: https://www.apachefriends.org/index.html
MAMP **Logotipo:** **Características:**	MAMP manage your website locally • Cuenta con todas las características de WAMP. • Se usa en sistemas Apple de allí la inicial M de las Macs. Su sitio oficial es: http://www.mamp.info/en/
BitNami **Logotipo:** **Características:**	bitnami • Colección de aplicaciones web como Wordpress, Drupal, Moodle, etc. • Cuenta con paquetes para cada aplicación, el cual puede ser instalado fácilmente además de poder virtualizarlo. Su sitio oficial es: https://bitnami.com/stacks
EasyPHP **Logotipo:** **Características:**	 • Tiene todas las características de WAMP pero orientada netamente a programadores PHP. • Tiene un control visible de los servidores y un registro de transacciones que almacena los sucesos del servidor. Su sitio oficial es: http://www.easyphp.org/

Sólo fines educativos - FreeLibros

NMP Server **Logotipo:** **Características:**	 - Es considerado un microservidor totalmente confiable pero principalmente portable; es decir, lo podrá ejecutar inclusive desde su unidad de almacenamiento externo. - Cuenta con un servidor análogo a Apache llamado nGinx. Su sitio oficial es: https://code.google.com/p/nmp-server/
UWAMP **Logotipo:** **Características:**	 - Cuenta con una interfaz bastante amigable. - Cuenta con una estadística gráfica del uso del procesador con respecto al servidor. Su sitio oficial es: http://www.uwamp.com/en/
Uniform Server **Logotipo:** **Características:**	 - Tiene todas las características ofrecidas por WAMP Server. - Es el servidor de menor tamaño comparado con sus análogos. Su sitio oficial es: http://www.uniformserver.com/
Softaculous AMPPS **Logotipo:** **Características:**	 - Tiene todas las características ofrecidas por WAMP Server. - Gestiona la base de datos MongoDB. - Usa los lenguajes de programación Python, Perl y PHP. - Gestiona la administración de archivos FTP. Su sitio oficial es: http://www.ampps.com/

2.5.1 Descargar WAMP Server

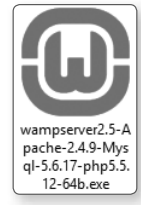
WAMP Server es una aplicación que no necesita licencia, por lo tanto, su descarga es totalmente libre. Ingresemos a la siguiente URL: www.wampserver.es

Fuente: <http://www.wampserver.es>

Sólo fines educativos - FreeLibros

Como podemos observar la página es bastante simple pero lo más importante es la facilidad que nos ofrece en la descarga del WAMP Server. Ahora nos ubicamos en la sección **DESCARGAR AHORA** y, según el tipo de sistema en el cual se desee instalar, deberá seleccionar la versión 64 o 32 bits.

Una vez finalizada la descarga obtendremos el archivo instalador, considere que para una instalación correcta debemos tener el archivo obtenido en una unidad estable de su computadora personal; es decir, no lo ejecute desde la unidad de almacenamiento portable, como los USB.



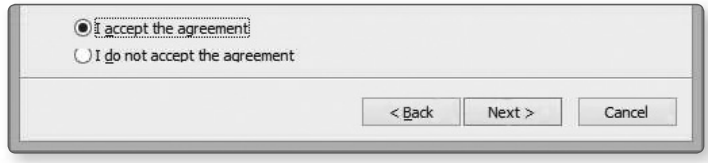
2.5.2 Instalación del servidor WAMP Server

Iniciaremos la instalación del servidor WAMP Server comentando que la instalación es muy sencilla de realizar, pues el instalador guiará en el proceso; además, hoy en día todo el mundo ha instalado alguna vez una aplicación en su computadora personal. Aun así, por motivos didácticos, a continuación mostraremos los siguientes pasos:

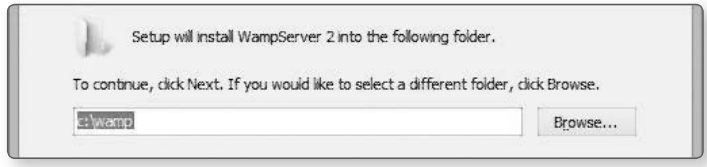
Paso 1: En el lado izquierdo de la ventana de bienvenida del instalador solo podremos observar las aplicaciones que se instalarán, seguidamente presionaremos el botón **Next>**.

Sólo fines educativos - FreeLibros

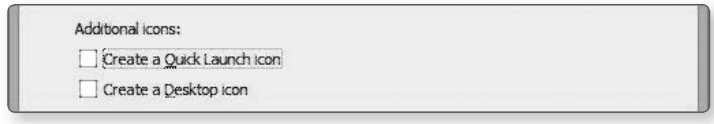
Paso 2: La ventana siguiente nos presenta la aceptación de la licencia, para lo cual solo debemos seleccionar la opción «I accept the agreement», seguidamente presionaremos el botón **Next>** para continuar con la instalación.



Paso 3: En la ventana siguiente debemos seleccionar la ubicación de la carpeta instalada; de forma predeterminada se presenta en la unidad **C:\wamp** y se recomienda que sea dicho lugar la ubicación de la instalación, seguidamente presionaremos el botón **Next>**.

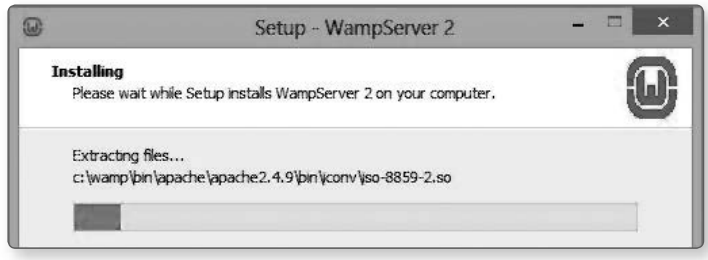


Paso 4: En la ventana siguiente podemos seleccionar la creación de íconos de acceso directo al servidor WAMP Server, tanto en el escritorio como en los íconos de acceso rápido, pero eso dependerá de usted, se sugiere no activarlo, seguidamente presionaremos el botón **Next>**.



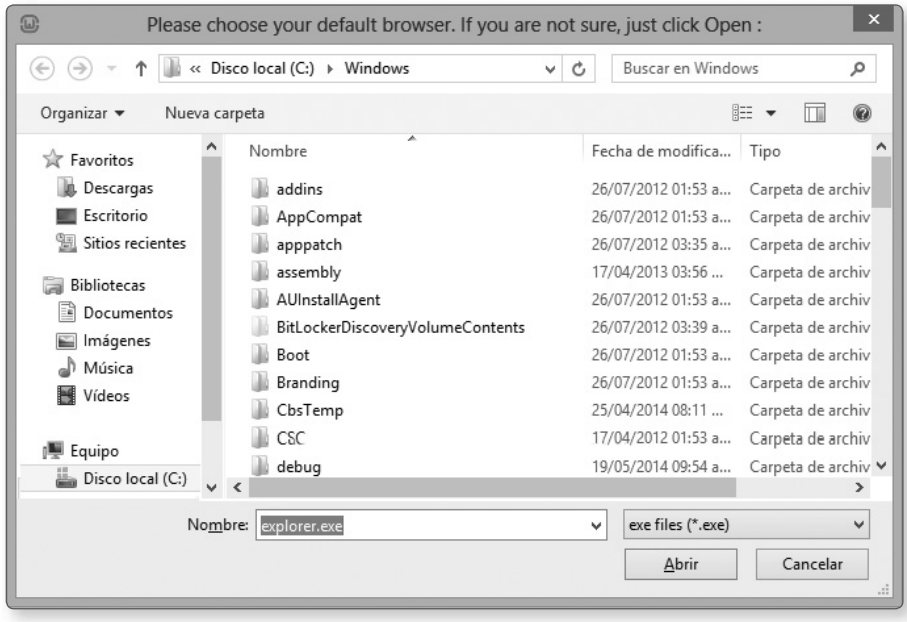
Paso 5: En la ventana siguiente se procederá a iniciar la instalación del servidor, para lo cual debemos presionar el botón **Install**.

Paso 6: En la ventana siguiente observaremos el proceso de instalación.

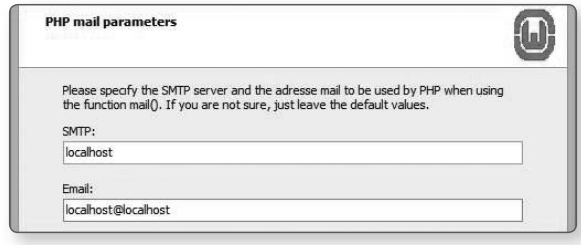


Paso 7: Antes de finalizar el proceso de instalación aparecerá una ventana de selección del navegador, en la cual se visualizarán las aplicaciones web. De forma predeterminada aparecerá seleccionado el **explorer.exe**, pero no se preocupe por eso pues cuando visualice sus documentos web podrá realizarlo desde cualquier navegador. Esta selección es solo una formalidad de la instalación.

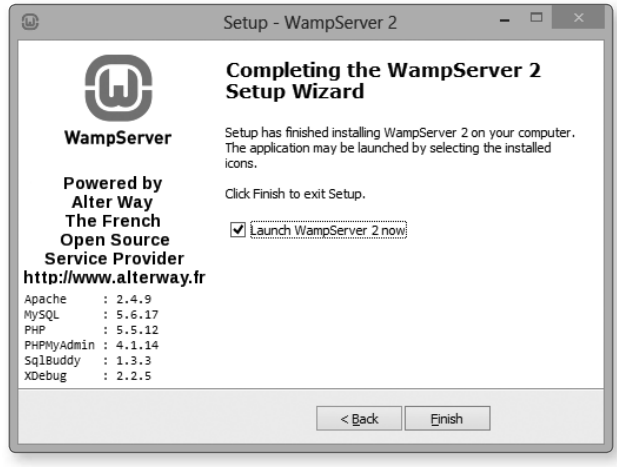
Sólo fines educativos - FreeLibros



Paso 8: En la ventana siguiente se ingresarán algunos parámetros PHP solicitados por el instalador, en la cual deberá configurar a SMTP con localhost y EMAIL con localhost@localhost, tal como se muestra en la imagen.



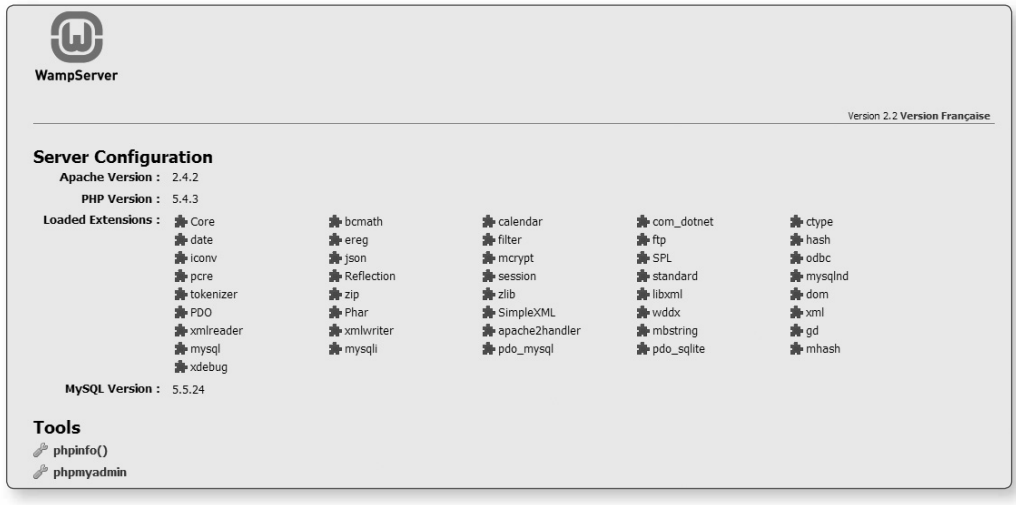
Paso 9: Si llegamos a la ventana «Completing the WampServer 2» entonces habremos completado la instalación correctamente. A continuación, procederemos a seleccionar el botón **Finish** para salir de la ventana; note que está activa la opción «Launch WampServer 2 now», que permitirá ejecutar la aplicación WampServer apenas cierre esta ventana.



Sólo fines educativos - FreeLibros

2.5.3 Pruebas del servidor WAMP

Una vez finalizada la instalación del WAMP Server debemos comprobar que las aplicaciones han sido instaladas correctamente, para esto debemos ingresar a su navegador web de preferencia y colocar en el URL la palabra **localhost**, si todo es correcto usted debe visualizar su navegador de la siguiente manera:



Fuente: <Imagen obtenida desde en el navegador Chrome>

2.5.4 Anomalías en la prueba del servidor WAMP

En ciertas ocasiones nos puede ocurrir que al realizar la prueba del servidor WAMP nos muestre la siguiente ventana, según su navegador:

Esta ventana nos indica que hay un error HTTP Error 404, pero antes de explicar el código del error mostrado veamos cómo funcionan las peticiones al servidor:

- El cliente realiza una petición.
- El servidor obtiene la dirección IP de la URL proporcionada.
- El servidor apertura una conexión de *socket* IP hacia la dirección IP.
- El servidor escribe un flujo de datos HTTP mediante el *socket* aperturado.
- El servidor recibe el flujo de datos HTTP desde el servidor web; este flujo de datos contiene códigos de estados cuyos valores son determinados por el protocolo HTTP y también muestra información encontrada, seguramente solicitada por el cliente.

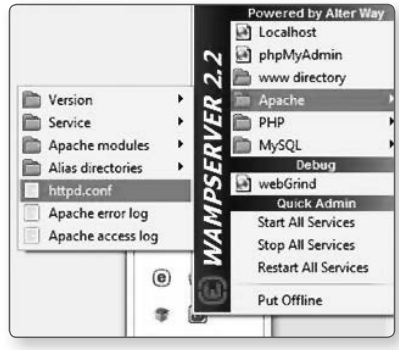
Sólo fines educativos - FreeLibros

Desde aquí se desprende el error HTTP 404, el cual indica que hubo un error en el flujo de datos desde el servidor; es decir, que no puede mostrar la información solicitada por el usuario. Eso puede ser porque en el servidor no encuentra dicho archivo web o porque el cliente se equivocó al escribir la URL.

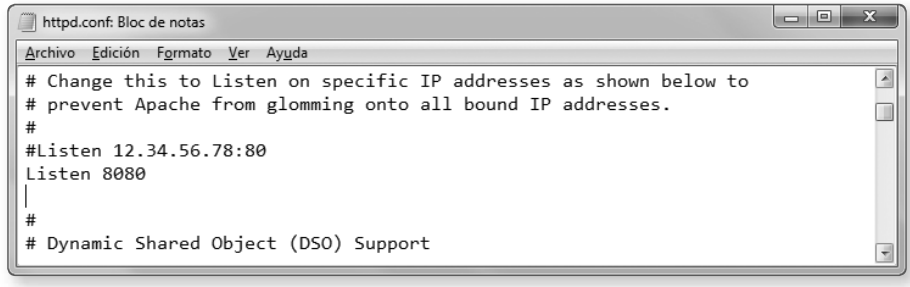
Otra forma de evidenciar que hay anomalías en el servidor es que el ícono presentado en los programas residentes en la memoria de la computadora se presenta de color naranja; en cualquiera de los casos podemos realizar los siguientes pasos para dar solución a dicha anomalía:

Pasos:

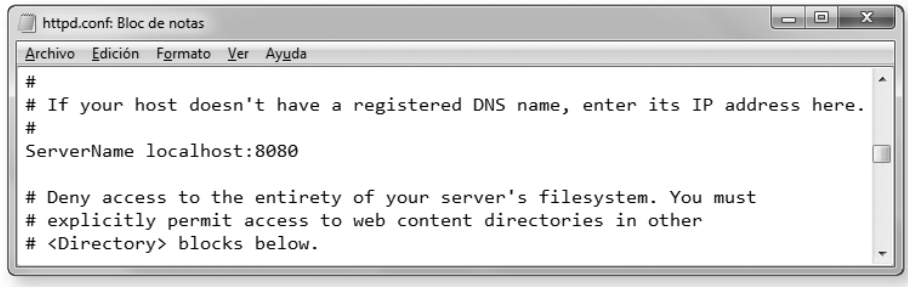
1. Desde el ícono WAMP Server, normalmente colocado al costado del reloj del sistema, seleccione **Apache > httpd.conf.**



2. Aparece la ventana editora desde la cual podrá realizar configuraciones que permitirán acceder al servidor WAMP Server de manera correcta; buscaremos la línea que contenga **Listen 80** y la modificaremos por **Listen 8080**, tal como se muestra en la siguiente imagen.

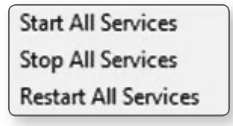


3. El cambio también lo haremos líneas más abajo, donde dice: **ServerName localhost:80** por **ServerName localhost:8080**, tal como se muestra en la siguiente imagen:



Sólo fines educativos - FreeLibros

4. Para que los cambios realizados se ejecuten correctamente en el servidor, debemos seleccionar la opciones **Start All Services** del ícono del WAMP Server, tal como se muestra en la siguiente imagen:



Finalmente, probaremos el servidor escribiendo la URL siguiente: **localhost:8080**. El resultado se muestra a continuación:

2.6 Instalación de Netbeans para PHP

Para el desarrollo de nuestras aplicaciones en PHP usaremos el IDE Netbeans, el cual se caracteriza por presentar una interfaz interactiva y por ser de fácil uso. Esto hace que nos preocupemos solo del código PHP, sobre el cual aprenderemos en los siguientes capítulos.

Si es la primera vez que instala Netbeans en su dispositivo, debe saber que necesita una plataforma de trabajo llamada JDK, el cual contiene todas las librerías necesarias para la ejecución de una aplicación PHP.

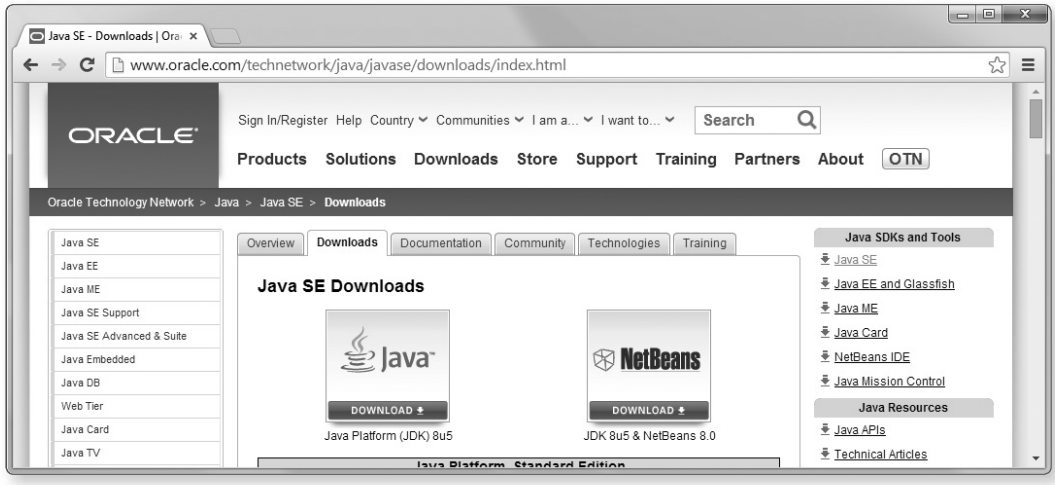
2.6.1 Paquete de aplicaciones JDK

JDK (Java Development Kit) incorpora un conjunto de aplicaciones y librerías que son necesarias para la ejecución de aplicaciones en Netbeans; por lo tanto, es lo primero que se debe ejecutar antes de instalar Netbeans.

Sólo fines educativos - FreeLibros

Pasos para la descarga e instalación de JDK:

Paso 1: Ingrese a la siguiente URL y seleccione la ficha **Downloads**, tal como se muestra en la siguiente imagen: http://www.oracle.com/technetwork/java/javase/downloads/index.html.

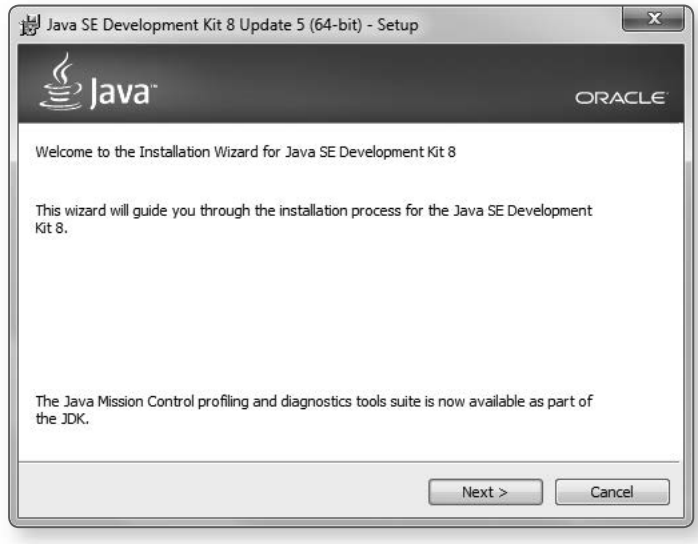


Fuente: <www.oracle.com>

Paso 2: En la sección de descarga seleccione el producto a descargar, dependiendo del sistema operativo con que cuenta:

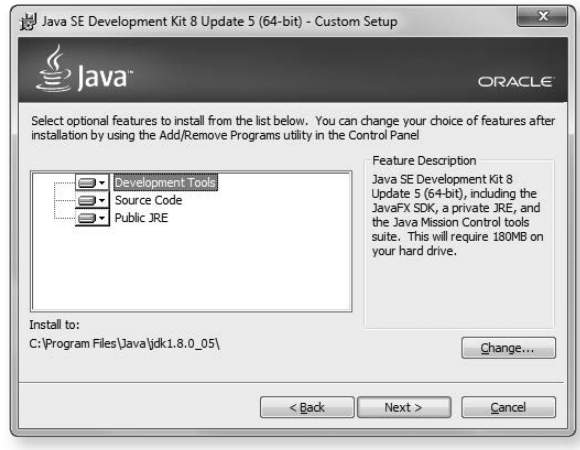
SISTEMA OPERATIVO	DESCARGAR
Windows 32 bits (Windows x86)	jdk-8u5-windows-i586.exe
Windows 64 bits (Windows x64)	jdk-8u5-windows-x64.exe

Paso 3: Procedemos con la instalación del JDK, a continuación le aparecerá la siguiente ventana, en la cual solo deberá presionar **Next>** ya que:



Sólo fines educativos - FreeLibros

Paso 4: Procedemos con la instalación del JDK, y aparecerá la siguiente ventana, en la cual presionaremos **Next>** para continuar con la instalación:



Paso 5: En todas las ventanas posteriores a la instalación se ha seleccionado el botón **Next>** hasta llegar a la ventana de «Instalación completa», en la cual presionaremos el botón **Close** para finalizar la instalación del JDK. Así, estaremos listos para la instalación de Netbeans. La siguiente imagen muestra la última ventana de la instalación:

2.6.2 IDE Netbeans

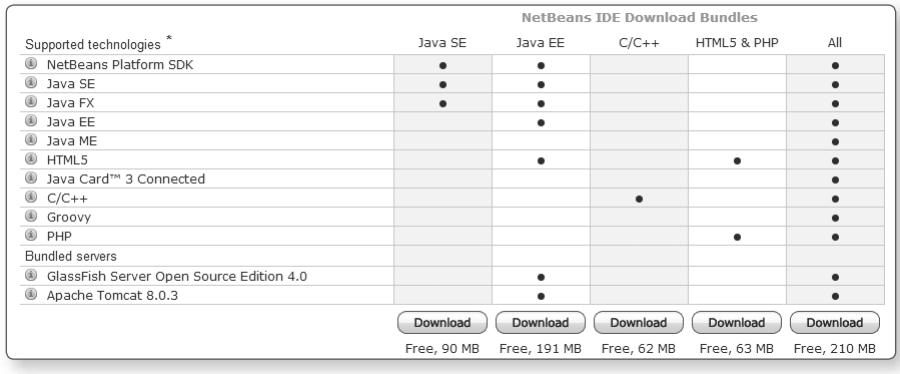
NetBeans es una aplicación desarrollada totalmente en un entorno libre, inicialmente fue desarrollado pensando en el lenguaje Java, hoy esa capacidad ha crecido de tal forma que se puede desarrollar códigos HTML5 y PHP dentro de la misma aplicación y con la seguridad de un producto libre y sin restricciones para su uso.

Fuente: <http://www.mundonets.com/herramientas/netbeans-ide/>

Está compuesto por librerías preparadas exclusivamente para el desarrollo de aplicaciones, como Java, PHP, HTML5, etc. De tal forma que cuando desarrolle sus aplicaciones en Netbeans, este le presentará las herramientas de trabajo en forma visual, evitando así codificar su entorno y dejando solo la preocupación en el código del lenguaje, en nuestro caso solo nos centraremos en la integración de HTML5 y PHP.

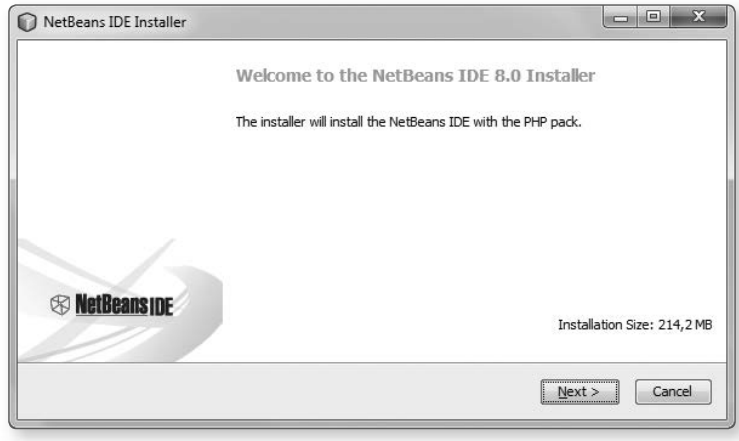
Sólo fines educativos - FreeLibros

Paso 1: Ingrese a la siguiente URL, tal como se muestra en la siguiente imagen: https://netbeans.org/downloads/index.html.

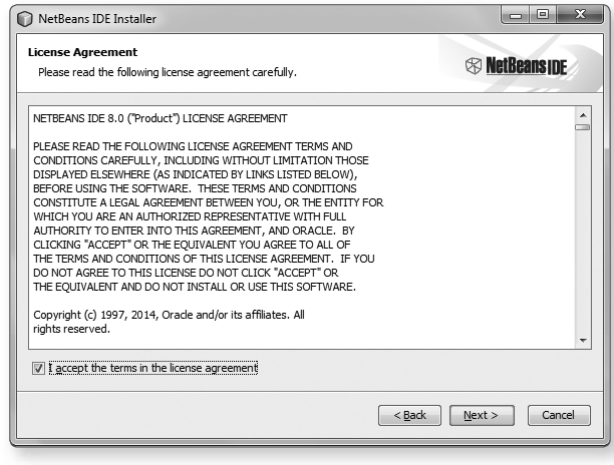


Seleccione el botón **Download** de la columna HTML5 & PHP, el cual tiene un peso de 63 MB; esta versión es la más ligera y contiene solo librerías para PHP con HTML5.

Paso 2: Una vez terminada la descarga de Netbeans 8.0 procedemos a la instalación del producto, seleccionando **Next>** de la siguiente ventana:

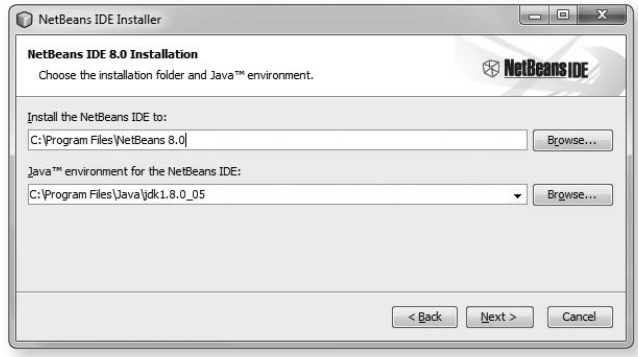


Paso 3: En la ventana posterior seleccionamos «I accept the terms the license agreement» y seguidamente el botón **Next>**, tal como se muestra en la siguiente imagen:



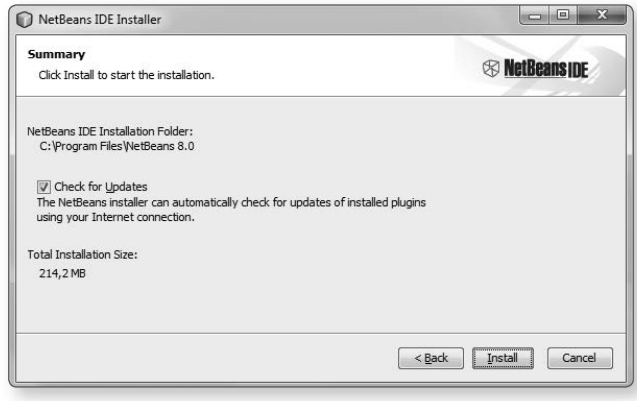
Sólo fines educativos - FreeLibros

Paso 4: La siguiente ventana comprueba que previamente haya sido instalado JDK en su computadora, si todo ha sido correcto solo presionamos **Next>**, tal como se muestra en la siguiente ventana:

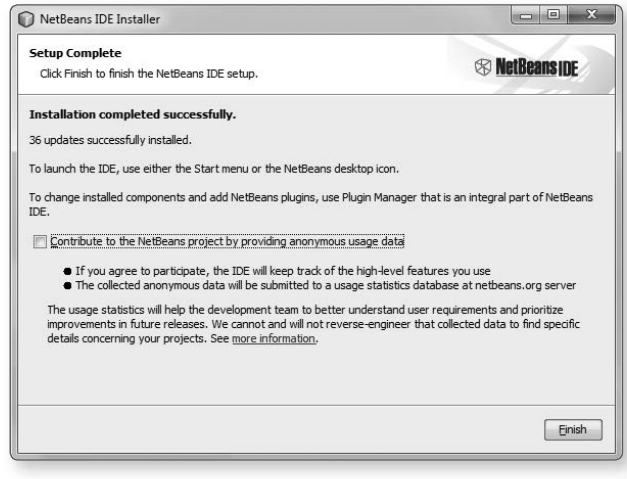


En caso de que esta ventana emita un error, puede que se deba a la versión del JDK; recuerde que debe instalar la versión del JDK perteneciente a la versión 8 del Netbeans.

Paso 5: En la siguiente ventana desactive la opción **Check for Updates** para las actualizaciones automáticas en el futuro, por parte de Netbeans; y para iniciar la instalación seleccione el botón **Install**, tal como se muestra en la siguiente imagen:



Paso 6: Una vez terminada la instalación, se procederá a desactivar la opciones **Contribute to the Netbeans…** a menos que usted opine lo contrario, para finalmente presionar el botón **Finish** y comenzar con el uso del IDE Netbeans, tal como se muestra en la siguiente imagen:



Sólo fines educativos - FreeLibros

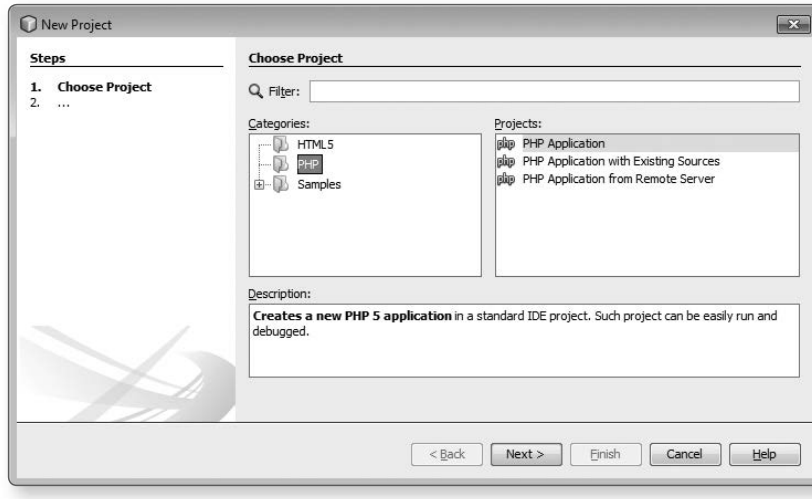
2.7 CUESTIONES POSTERIORES A LA INSTALACIÓN DEL NETBEANS

Nombraremos las principales cuestiones posteriores la instalación del Netbeans, como crear nuestro primer proyecto en PHP o cambiar el tipo de letra del código, etc. Tomaremos como escenario inicial el hecho de que tenemos levantado el servidor WAMP Server en nuestra computadora, configurada en el puerto 8080.

- **Cuestión 1: Crear un proyecto en Netbeans**

Paso 1: Desde el menú de opciones seleccione **File > New Project.**

Paso 2: Desde la ventana «New Project» seleccione el tipo de aplicación a desarrollar. Para nuestro caso seleccionaremos «PHP», luego el proyecto «PHP Application» y finalmente **Next >**, tal como se muestra en la siguiente imagen:

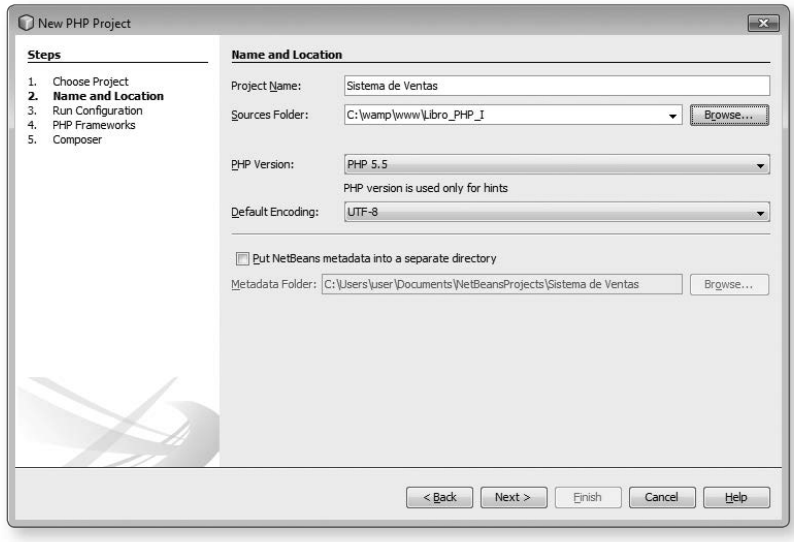


Paso 3: Desde la ventana «New PHP Project» ingrese el nombre del proyecto en Project Name, luego defina el destino de la aplicación; es decir, el lugar desde el cual se ejecutará, para lo cual debe seleccionar el botón **Browse...** y direccionar según el caso:

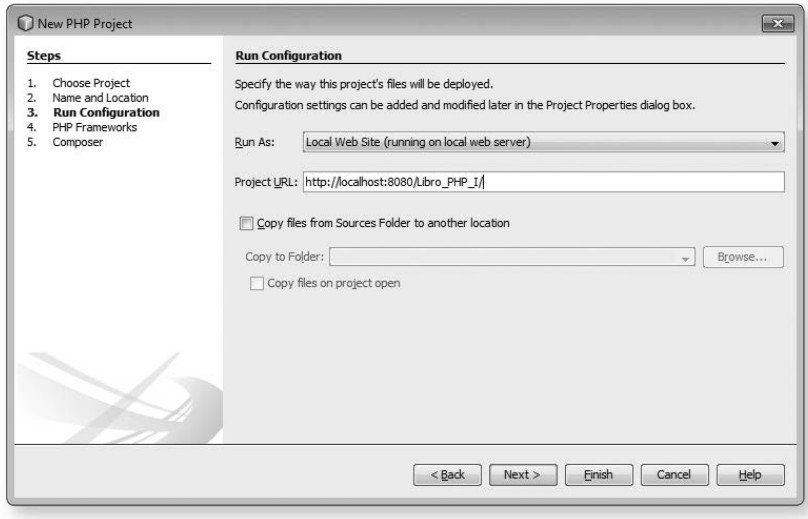
SERVIDOR	CARPETA DESTINO
WAMP Server	C:\wamp\www
XAMP	C:\xamp\httpdocs

Se recomienda crear una carpeta para una mejor organización de los proyectos dentro de la carpeta **www**; en este caso se ha creado una carpeta llamada «Libro_PHP_I». Finalmente, se debe presionar el botón **Next >**, tal como se muestra en la siguiente ventana:

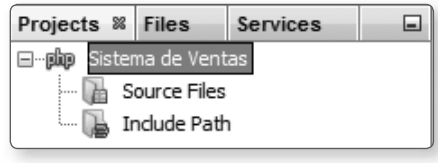
Sólo fines educativos - FreeLibros



Paso 4: En la siguiente ventana asegúrese de que la dirección Project URL sea la correcta, pues cuando ejecute una aplicación PHP esta dirección mostrará el resultado. Para nuestro caso hemos agregado 8080 a la URL, pues así se configuró el Wamp Server; si usted no hizo cambios se sugiere no modificar nada de esta ventana. Finalmente, presione **Finish** para crear el proyecto tal como se muestra en la siguiente ventana:



Finalmente, la paleta Project se mostrará de la siguiente manera:

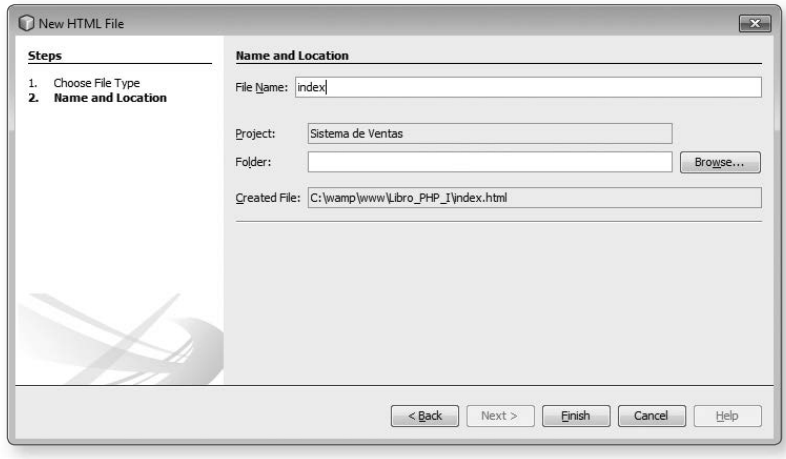


Sólo fines educativos - FreeLibros

Donde la carpeta **Source Files** representa la fuente de la aplicación; es decir, aquí se almacenarán todos los archivos web, como HTML, PHP o CSS de la aplicación. Es aquí donde nos concentraremos cuando necesitemos visualizar el contenido de algún archivo.

Cuestión 2: Agregar un archivo HTML5 al proyecto

Paso 1: Desde la paleta Projects presionaremos clic derecho sobre **Source Files** y seleccionaremos **New > HTML File…**



Paso 2: Desde la ventana «New HTML File» ingrese el nombre del archivo, para nuestro caso le asignamos **index**, en minúsculas, y finalmente presione el botón **Finish** de tal manera que se presente el siguiente script:

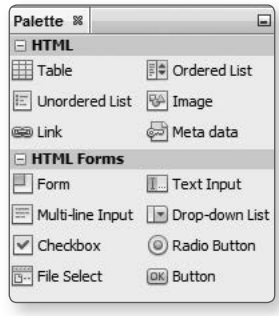
```
<!DOCTYPE html>
<!--
To change this license header, choose License Headers in Project Properties.
To change this template file, choose Tools | Templates
and open the template in the editor.
-->
<html>
    <head>
        <title>TODO supply a title</title>
        <meta charset="UTF-8">
        <meta name="viewport" content="width=device-width, initial-scale=1.0">
    </head>
    <body>
        <div>TODO write content</div>
    </body>
</html>
```

Mientras que la paleta Projects muestra el siguiente aspecto:

Sólo fines educativos - FreeLibros

- **Cuestión 3: Agregar la paleta con etiquetas HTML5**

Paso 1: Desde el menú de opciones seleccione **Window > IDE Tools > Pallete.**



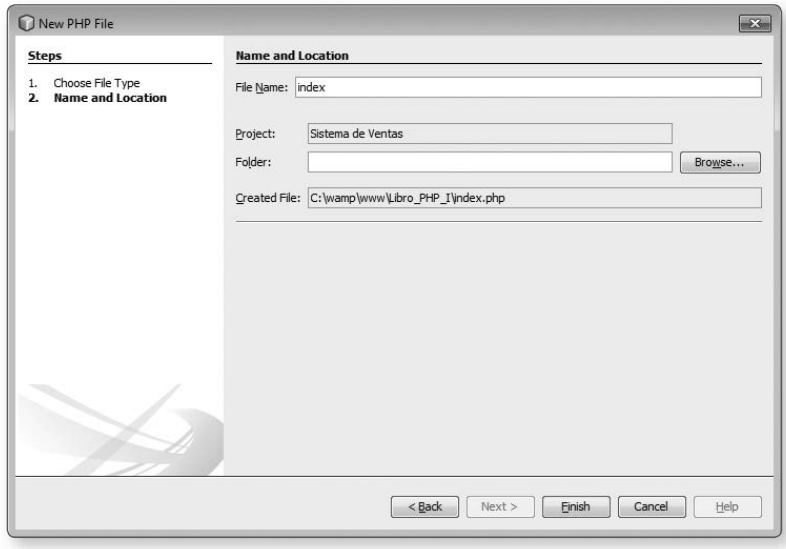
Paso 2: Para insertar un elemento HTML al proyecto tenemos dos posibilidades:

- Ubicar el cursor donde insertará la etiqueta, luego presione doble clic sobre la etiqueta.
- Arrastrar la etiqueta desde la palette hacia el script, y soltarlo en el lugar donde se escribirá el código.

- **Cuestión 4: Agregar y ejecutar un archivo PHP**

El agregar un archivo PHP permite incorporar un archivo al proyecto liberado de código HTML5.

Paso 1: Desde la paleta Projects presionaremos clic derecho sobre **Source Files** y seleccionaremos **New > PHP File…**



Sólo fines educativos - FreeLibros

Paso 2: Desde la ventana anterior solo seleccionaremos el botón **Finish**, puesto que todo ha sido configurado en el proyecto inicial.

```
<?php

/*
 * To change this license header, choose License Headers in Project Properties.
 * To change this template file, choose Tools | Templates
 * and open the template in the editor.
 */
```

Desde aquí podemos modificar nuestro código PHP borrando los comentarios y colocando el siguiente script:

```
<?php
    echo 'Hola mundo con PHP y Netbeans';
?>
```

Finalmente, para ejecutar la aplicación presionaremos la combinación de teclas:

SHIFT + F6

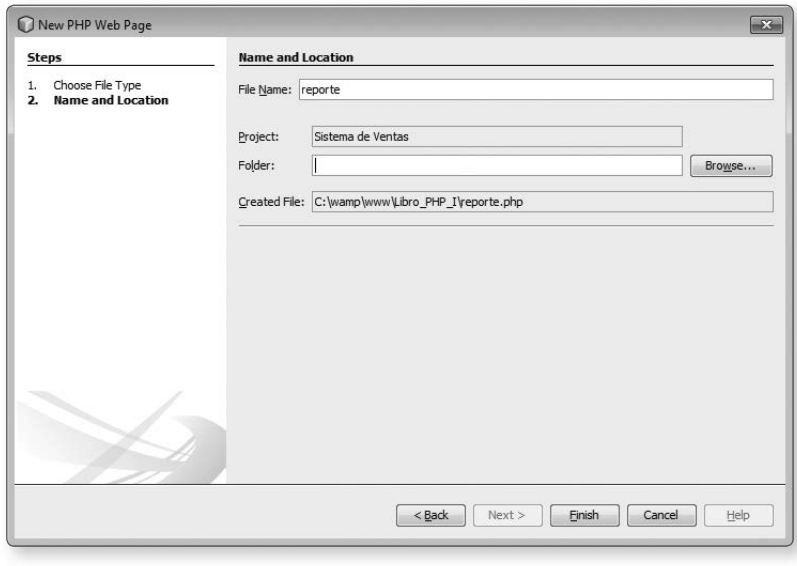
Emitiendo el siguiente resultado en el navegador por defecto:

Hola mundo con PHP y Netbeans

- **Cuestión 5: Agregar y ejecutar un archivo de página web PHP**

El agregar un archivo de página web PHP permite incorporar un archivo HTML5 integrado con código PHP al proyecto.

Paso 1: Desde la paleta Projects presionaremos clic derecho sobre **Source Files** y seleccionaremos **New > PHP Web Page...**



Sólo fines educativos - FreeLibros

Paso 2: Desde la ventana anterior seleccionaremos el botón **Finish**, puesto que todo ha sido configurado al iniciar el proyecto.

```
<!DOCTYPE html>
<!--
To change this license header, choose License Headers in Project Properties.
To change this template file, choose Tools | Templates
and open the template in the editor.
-->
<html>
    <head>
        <meta charset="UTF-8">
        <title></title>
    </head>
    <body>
        <?php
        // put your code here
        ?>
    </body>
</html>
```

Desde aquí podemos modificar nuestro código borrando los comentarios y colocando el siguiente script:

```
<!DOCTYPE html>
<html>
    <head>
        <meta charset="UTF-8">
        <title></title>
    </head>
    <body>
        <?php
            echo 'Hola mundo con HTML5 y PHP';
        ?>
    </body>
</html>
```

Finalmente para ejecutar la aplicación presionaremos la combinación de teclas:

SHIFT + F6

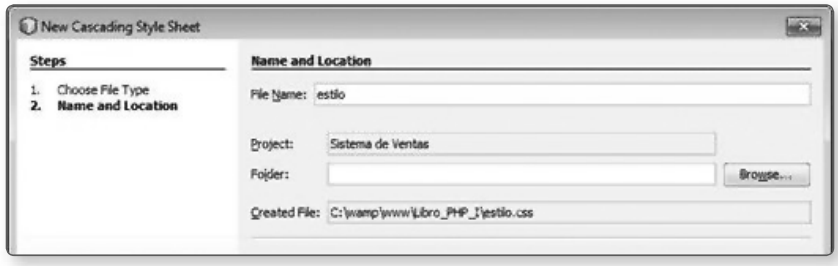
Emitiendo el siguiente resultado en el navegador por defecto:

- **Cuestión 6: Agregar un archivo CSS al proyecto**

Los archivos CSS son necesarios dentro de un proyecto PHP, porque le dará todos los formatos que necesiten, tanto el código HTML5 como el PHP. Para incorporar un archivo CSS realizaremos los siguientes pasos:

Paso 1: Desde la paleta Projects presionaremos clic derecho sobre **Source Files** y seleccionaremos **New > Cascanding Style Sheet…**

Sólo fines educativos - FreeLibros



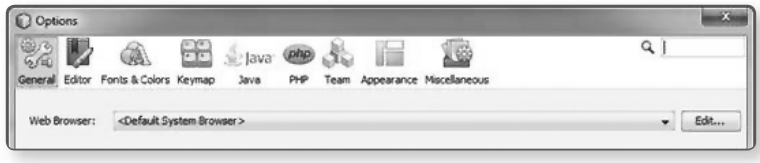
Paso 2: De la ventana anterior solo asignaremos el nombre del estilo y presionaremos **Finish,** al principio se mostrará el siguiente script, el cual usted modificará y agregará los estilos necesarios para su aplicación:

```
/*
To change this license header, choose License Headers in Project Properties.
To change this template file, choose Tools | Templates
and open the template in the editor.
*/
/*
    Created on : 28/07/2014, 10:49:13 AM
    Author     : user
*/
```

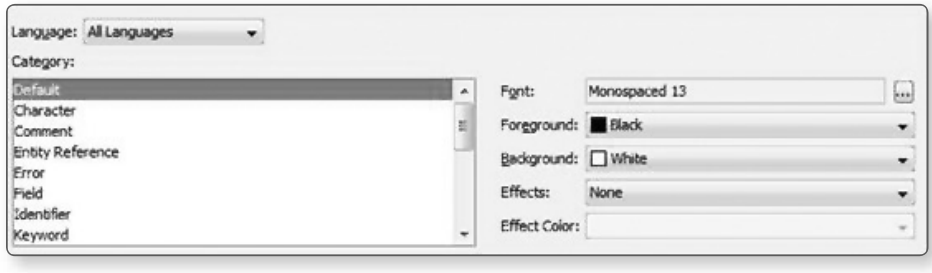
- **Cuestión 7: Modificar la fuente y tamaño del código mostrado en el editor de Netbeans**

Se recomienda visualizar el tipo de fuente **Consolas** en el editor de código, tanto para PHP como para HTML5. Para realizar dicho cambio realizaremos los siguientes pasos:

Paso 1: Desde el menú de opciones seleccione **Tools > Options.**



Paso 2: Desde la ventana anterior seleccione la opción **Fonts & Colors.**



Sólo fines educativos - FreeLibros

Paso 3: Seleccione desde la categoría la opción **Default**, luego presione «…» desde la opción **Font** y seleccione **Consolas** con el tamaño que crea conveniente, tal como se muestra en la siguiente ventana:

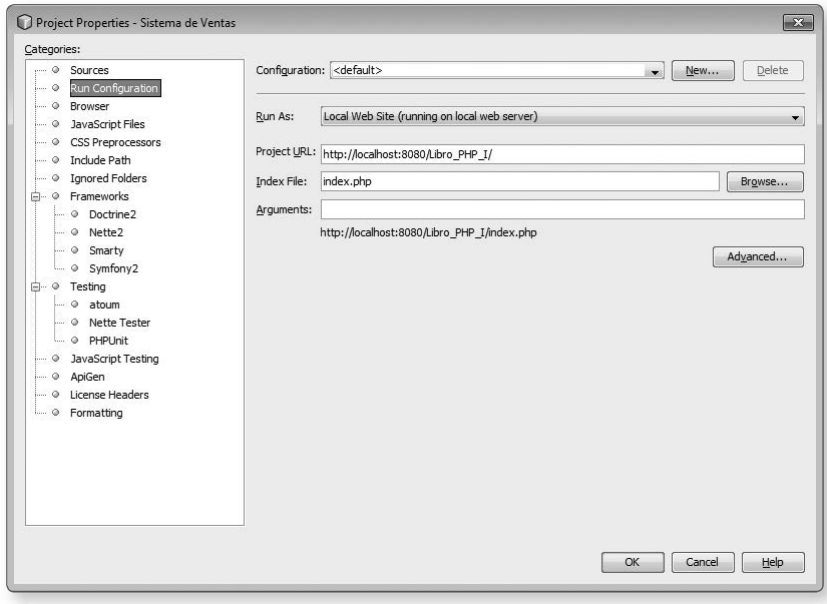
Cuestión 8: Modificar las ubicaciones de los proyectos al ejecutarlos

Cuando se requiere ejecutar un proyecto en otra computadora debemos seguir los siguientes pasos:

Paso 1: Copiar la carpeta del proyecto en el servidor actual de la computadora, recuerde que de acuerdo al servidor la carpeta destino varía, tal como se muestra en el siguiente cuadro:

SERVIDOR	CARPETA DESTINO
WAMP Server	C:\wamp\www
XAMP	C:\xamp\httpdocs

Paso 2: Ahora procederemos a modificar la ubicación de los proyectos para poder ejecutarlos, presione clic derecho sobre el nombre del proyecto ubicado en la paleta Projects y seleccione Properties…



Sólo fines educativos - FreeLibros

De la ventana anterior seleccione la categoría **Run Configuration** y modifique la opción Project URL:

Cuestión 9: Modificar el navegador predeterminado

Cuando se ejecuta una aplicación con la combinación de teclas Shift+F6 el resultado se muestra en un navegador predeterminado por Netbeans. Si usted tiene un navegador en especial, entonces necesitará modificarlo de la siguiente manera:

Paso 1: Presionaremos clic derecho sobre el proyecto que se ubica en la paleta Projects, y seleccionaremos Properties, tal como se muestra en la siguiente imagen:

Paso 2: De la ventana anterior seleccione la categoría **Browser** y seleccione su navegador preferido, tenga en cuenta que la próxima vez que ejecute su aplicación se verá reflejada en el navegador predeterminado.

Sólo fines educativos - FreeLibros

Sólo fines educativos - FreeLibros

CAP.

3

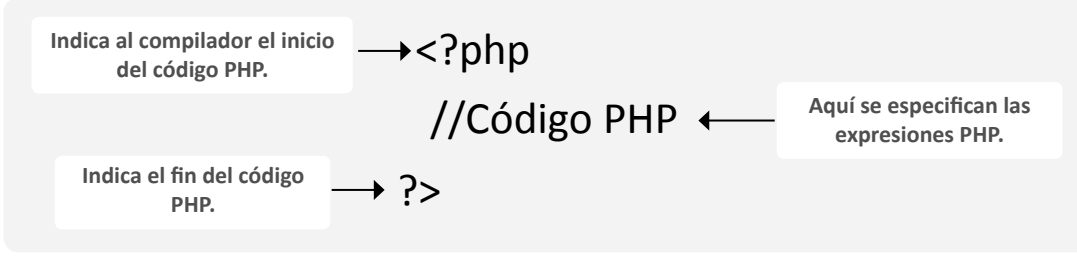
Lenguaje PHP

Sólo fines educativos - FreeLibros

Sólo fines educativos - FreeLibros

3.1 INTEGRAR CÓDIGO PHP EN HTML5

PHP por sí solo puede realizar gestionar muchas de las tareas que realiza HTML5, pero el resultado no tendría el mismo aspecto pues PHP es un lenguaje de programación; por lo tanto, debemos entender que la integración entre PHP y HTML5 se debe simplemente a temas de diseño. Veamos el formato de integración de PHP en un documento HTML5:



Una segunda forma de integrar código PHP en HTML5 sería usando el siguiente formato:

```
<script language="PHP">
    //Código PHP
</script>
```

Veamos el siguiente ejemplo de integración entre código PHP y HTML5:

Tener en cuenta:

- Se debe usar etiquetas HTML5.
- La página debe dividirse en tres secciones: título (**Header**), contenido (**Section**) y pie (**Footer**).
- Debe enlazar a un archivo **estilo.css** que contenga estilos básicos para presentar el documento de la mejor manera posible.

Sólo fines educativos - FreeLibros

Archivo: **integración.php**

```
<!DOCTYPE html>
<html>
    <head>
        <meta charset="UTF-8">
        <link href="estilo.css" rel="stylesheet">
        <title>Integrar PHP con HTML5</title>
    </head>
    <body>
        <header>
            <h3>Capitulo 3: Fundamentos de Programación con PHP</h3>
        </header>
        <section>
            <?php
                echo "<br>3.1 Integrar código PHP en HTML5";
                echo "<br>3.2 Página estática versus página dinámica";
                echo "<br>3.3 Manejo de literales de programación";
                echo "<br>3.4 Manejo de operadores y expresiones";
                echo "<br>3.5 Salida de información con PHP";
                echo "<br>3.6 Manejo de variables";
                echo "<br>3.7 Tipos de datos usados en PHP";
                echo "<br>3.8 Manejo de constantes";
                echo "<br>3.9 Casos desarrollados";
            ?>
        </section>
        <footer>
            <h6>Todos los derechos reservados - Lic. Manuel Torres</h6>
        </footer>
    </body>
</html>
```

También se implementó un archivo CSS que permitiera formatear el tipo de letra y tamaño de los textos en la impresión:

Archivo: **estilo.css**

```
body{
    font-family: "tahoma";
    font-size: 14px;
}
```

3.2 Salida de información con PHP

La salida de información resulta muy importante para PHP, pues desde aquí se podrá enviar una respuesta al cliente, veamos dos alternativas para la salida de información:

3.2.1 Función echo

Es la función más representativa para PHP, su formato es el siguiente:

echo "mensaje"

Sólo fines educativos - FreeLibros

Se pueden presentar las siguientes características de la función **echo**:

- Se puede usar un juego de comillas simples (' ') o dobles (" ") para un determinado mensaje.
- Un mensaje podría ser una etiqueta del HTML5.
- Cuando se imprima un juego de mensajes y variables se debe concatenar usando el operador punto (.).
- Para finalizar correctamente una función **echo** se asigna un punto y coma (;) al final de la línea.

Veamos algunos ejemplos:

```
echo "El sueldo básico es: ";
echo "<h3>El sueldo básico es:</h3>";
echo "El sueldo básico es: ".$sueldoBasico;
```

3.2.2 Función printf

Permite la impresión en el navegador de manera formateada, la sintaxis es la siguiente:

printf("Mensaje");
printf("Mensaje",variable1,..VariableN);

Se pueden presentar las siguientes características de la función **printf**:

- Puede imprimir mensajes simples e inclusive combinarlos con etiquetas HTML5.
- Para encerrar los mensajes puede usar el juego de comillas simples o dobles.
- La impresión de una variable conlleva a la especificación del tipo de salida, estas son:
  - **%s**: Para impresión de cadena de caracteres.
  - **%d**: Para impresión de números que no incluyan decimales.
  - **%f**: Para impresión de números que incluyan decimales.

Veamos algunos ejemplos:

```
printf("El sueldo básico es: ");
printf("<h3>El sueldo básico es:</h3>");

$sueldoBasico=350;
printf("El sueldo básico es: %.2f",$sueldoBasico);
```

Sólo fines educativos - FreeLibros

3.2.3 Comentarios PHP

Los comentarios en un código PHP pueden tener diferentes perfiles, uno de ellos sirve comentar sobre una sección del código o mostrar un mensaje al usuario; en otro caso se podría usar para anular una parte del código PHP, ya que los comentarios no son reconocidos por el intérprete de PHP.

Tenemos los siguientes formatos:

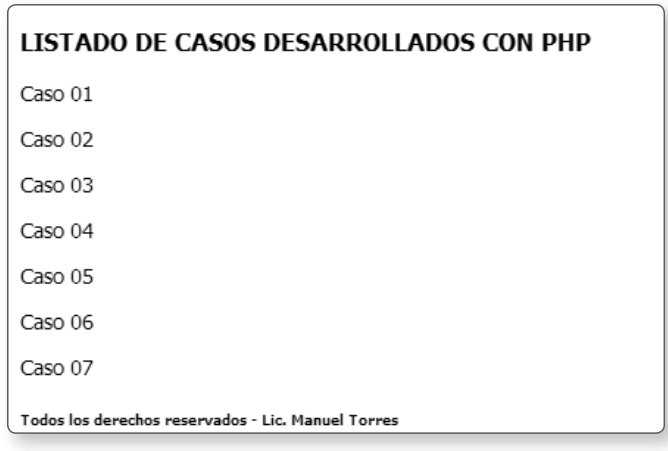
Operador	Descripción

//	Permite comentar una sola línea.
	
	`<?php`
	`  //Captura de valores`
	`  $n=$_POST["txtN"];`
	
	`  //Impresión de valor`
	`  echo $n;`
	`?>`

/* */	Permite comentar un conjunto de líneas.
	
	`<?php`
	`  /*`
	`  Lenguaje: PHP`
	`  Autor: Lic. Manuel Torres`
	`  Fecha: Junio 2015`
	`  */`
	`  $n=$_POST["txtN"];`
	`  echo $n;`
	`?>`

3.3 Página estática versus página dinámica

Una página estática es desarrollada principalmente por HTML5. A continuación veremos cómo implementar el siguiente caso:



Sólo fines educativos - FreeLibros

Archivo: **index.php**

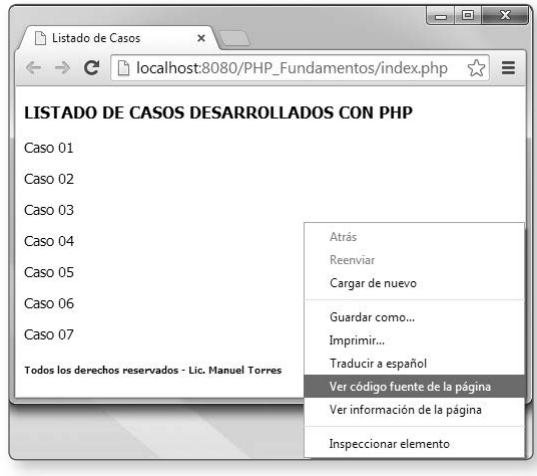
```
<!DOCTYPE html>
<html>
    <head>
        <meta charset="UTF-8">
        <link href="estilo.css" rel="stylesheet">
        <title>Listado de Casos</title>
    </head>
    <body>
        <header>
            <h3>LISTADO DE CASOS DESARROLLADOS CON PHP</h3>
        </header>
        <section>
            <p>Caso 01</p>
            <p>Caso 02</p>
            <p>Caso 03</p>
            <p>Caso 04</p>
            <p>Caso 05</p>
            <p>Caso 06</p>
            <p>Caso 07</p>
        </section>
        <footer>
           <h6>Todos los derechos reservados - Lic. Manuel Torres</h6>
        </footer>
    </body>
</html>
```

También se implementó un archivo CSS que permita formatear el tipo de letra y tamaño de los textos en la impresión:

Archivo: **estilo.css**

```
body{
    font-family: "tahoma";
    font-size: 14px;
}
```

Como notará, la implementación se debe toda vez al uso del código HTML5. Ahora veremos qué sucede si el usuario decide analizar el código de dicha página web:



Sólo fines educativos - FreeLibros

Para ver el código fuente de la página web, tenemos que presionar clic derecho sobre el fondo de la página web y seleccionar «Ver código fuente de la página»; el resultado es el siguiente:

```
<!DOCTYPE html>
<html>
    <head>
        <meta charset="UTF-8">
        <link href="estilo.css" rel="stylesheet">
        <title>Listado de Casos</title>
    </head>
    <body>
        <header>
            <h3>LISTADO DE CASOS DESARROLLADOS CON PHP</h3>
        </header>
        <section>
            <p>Caso 01</p>
            <p>Caso 02</p>
            <p>Caso 03</p>
            <p>Caso 04</p>
            <p>Caso 05</p>
            <p>Caso 06</p>
            <p>Caso 07</p>
        </section>
        <footer>
           <h6>Todos los derechos reservados - Lic. Manuel Torres</h6>
        </footer>
    </body>
</html>
```

Como podemos observar, el mismo código que implementó a la página web se manifiesta frente al usuario.

Ahora veremos la implementación del mismo caso pero con código PHP, el resultado normalmente es el mismo, pero ya no es considerado como una página estática sino, más bien, como una página dinámica. Veamos el código:

Archivo: **dinamico.php**

```
<!DOCTYPE html>
<html>
    <head>
        <meta charset="UTF-8">
        <link href="estilo.css" rel="stylesheet">
        <title>Listado de Casos</title>
    </head>
    <body>
        <header>
            <h3>LISTADO DE CASOS DESARROLLADOS CON PHP</h3>
        </header>
        <section>
            <?php
                for($i=1;$i<=7;$i++){
            ?>
            <p>Caso <?php echo $i; ?> </p>
            <?php
                }
            ?>
        </section>
        <footer>
           <h6>Todos los derechos reservados-Lic. Manuel Torres</h6>
        </footer>
    </body>
</html>
```

Sólo fines educativos - FreeLibros

Y si analizamos el código fuente de la página generada a partir del código dinámico; esta será la imagen que el usuario visualizará:

```
<!DOCTYPE html>
<html>
    <head>
        <meta charset="UTF-8">
        <link href="estilo.css" rel="stylesheet">
        <title>Listado de Casos</title>
    </head>
    <body>
        <header>
            <h3>LISTADO DE CASOS DESARROLLADOS CON PHP</h3>
        </header>
        <section>
                        <p>Caso 1 </p>
                        <p>Caso 2 </p>
                        <p>Caso 3 </p>
                        <p>Caso 4 </p>
                        <p>Caso 5 </p>
                        <p>Caso 6 </p>
                        <p>Caso 7 </p>
                    </section>
        <footer>
           <h6>Todos los derechos reservados-Lic. Manuel Torres</h6>
        </footer>
    </body>
</html>
```

Como notará, el usuario no podrá visualizar el código PHP que fue insertado en el HTML5, para esto el intérprete convierte las instrucciones PHP en código HTML5 entendible por el cliente, y es así como manejaremos todas las aplicaciones PHP de aquí en adelante.

3.4 MANEJO DE LITERALES DE PROGRAMACIÓN

Los literales en PHP son considerados como valores que puede asumir una determinada variable; dichos valores son de diferentes tipos, tal como veremos a continuación:

LITERALES PHP	VALORES
Enteros	10, 100, 1000, 10000, -10, -100, -1000
Reales	20.50, 0.0225352, -20.50, -0.0225352
Booleano	true, false
Carácter	'a', 'b' o también "a", "b"
Cadena de caracteres	'Jefe', 'Administrador' o también "Jefe", "Administrador"

A diferencia de otros lenguajes de programación, PHP toma el juego de comillas simples y dobles de la misma manera; es decir, cuando se haga referencia a un literal carácter o de cadena deberá estar encerrado entre comillas, ya sean simples o dobles, como se visualiza en el cuadro anterior.

3.5 MANEJO DE OPERADORES

Cuando se implementan aplicaciones PHP se tendrá la necesidad de usar operadores de todo tipo, por ejemplo, en las condicionales necesitaremos operadores de comparación y relacionales. Aquí veremos una lista de los operadores más importantes que presenta PHP.

Sólo fines educativos - FreeLibros

3.5.1 Operadores aritméticos

Permiten armar una expresión o sentencia PHP que dará un resultado esperado al usuario.

Operador	Descripción	Ejemplo
+	Permite sumar dos o más variables de tipo numérico.	$s = $n1 + $n2;
-	Permite restar dos variables de tipo numérico, pero también puede representar la negación de un valor.	$r = $n1 - $n2; $neg = -$n1;
*	Permite multiplicar dos o más variables de tipo numérico.	$m = $n1 * $n2;
/	Permite dividir entre dos elementos de tipo numérico.	$d = $n1 / $n2;
%	Permite obtener el residuo de dividir dos variables de tipo numérico.	$res = $n1 % $n2;

3.5.2 Operadores de cadena de caracteres

Permite unir dos o más caracteres formando así una cadena concatenada.

Operador	Descripción	Ejemplo
.	Operador de concatenación, el cual permite unir dos o más valores, no necesariamente del mismo tipo.	$concatenar = $n." "; echo "El número es: ".$n;

3.5.3 Orden de prioridad de los operadores

Debemos considerar que el orden de los elementos es muy importante al momento de crear nuestras expresiones de cálculos en PHP, considere la siguiente tabla:

Orden de prioridad	Operador
1	++ --
2	Casting (int) (float) (string) (bool) (object)
3	!
4	* / %
5	+ -
6	< <= > >=
7	== != ===
8	&&
9	||
10	= += -= *= /=

Los operadores nuevos que aparecen en la tabla anterior se conceptualizarán en los capítulos posteriores.

Sólo fines educativos - FreeLibros

3.6 MANEJO DE VARIABLES

Una variable en PHP, así como en otros lenguajes de programación, constan de dos partes: la primera es el nombre de la variable; la otra, el contenido de valor que tendrá la variable, por ejemplo **$n=10;** **$n** es la variable mientras que 10 es el valor de la variable.

En PHP no será necesario especificar el tipo de datos cuando se declare una variable, esta será declarada implícitamente por el valor que se le asigne en tiempo de ejecución; es decir, cuando el usuario ingrese un valor, automáticamente se declarará como tipo del dato asignado.

Consideraciones para la declaración de una variable:

- Toda variable de PHP inicia con el símbolo dólar ($).
- El primer carácter después del símbolo dólar tiene que ser obligatoriamente una letra.
- Los demás caracteres pueden ser letras, números o la combinación de ellos.
- La variable de PHP es sensible a la mayúscula y minúscula; por lo tanto, se recomienda digitarlo totalmente en minúsculas.

El formato para la declaración de la variable PHP es:

$variable = valor;

Veamos algunos ejemplos de declaración de variables:

DECLARACIÓN	DESCRIPCIÓN
$n=10	Declarando la variable **n** de tipo entero, con un valor asignado de 10.
$sueldo=4000.60	Declarando la variable **sueldo** de tipo decimal, con un valor inicial de 4000.60.
$empleado="Ángela Torres"	Declarando la variable **empleado** de tipo cadena, con un valor inicial de “Ángela Torres”.
$estadoCivil="s"	Declarando la variable **estado civil** de tipo cadena, con un valor inicial de “s”.
$a=$b;	Declarando la variable **a** del mismo tipo que la variable **b**.

También podemos declarar una variable con una expresión de la siguiente manera:

DECLARACIÓN	DESCRIPCIÓN
$r = $n+pow($a,2)	Declarando la variable **r**, donde el tipo de datos dependerá del resultado de la expresión $n+pow ($a,2), si esta resulta ser entera o decimal.
$c = "Resultado".$r;	Declarando la variable **c**, donde el resultado es de tipo cadena pues inicialmente presenta el texto “Resultado” concatenado con la variable $r.
$f = date("d-m-Y H:i:s");	Declarando la variable **f** de tipo cadena, pues almacena la fecha actual y es tomada así, porque PHP no cuenta con un tipo de datos para fechas.

Sólo fines educativos - FreeLibros

3.7 Tipos de datos usados en PHP

Nombraremos las novedades de la versión estable que presenta PHP hasta el día de la elaboración de este material; es decir, revisaremos la versión 5.4.3 del PHP.

A continuación, visualizaremos los tipos de datos de un trabajador a partir de sus datos personales.

Archivo: **tipos.php**

```
<!DOCTYPE html>
<html>
<head>
<meta charset="UTF-8">
<link href="estilo.css" rel="stylesheet">
<title>Listado de Casos</title>
</head>
  <body>
   <?php
    $trabajador='Fernanda Ximena Torres Lázaro';
    $fechaNac='10/10/1985';
    $numeroHijos=2;
    $sueldo=4500.10;
    $estado=true;

    echo 'Valor de la variables Tipo de Datos';
    echo '<br>-----------------------------------------------------';
    echo '<br>'.$trabajador.' '.gettype($trabajador);
    echo '<br>'.$fechaNac. ' '.gettype($fechaNac);
    echo '<br>'.$numeroHijos. ' '.gettype($numeroHijos);
    echo '<br>'.$sueldo. ' '.gettype($sueldo);
    echo '<br>'.$estado. ' '.gettype($estado);
   ?>
 </body>
</html>
```

Comentarios:

```
$trabajador='Fernanda Ximena Torres Lázaro';
$fechaNac='10/10/1985';
$numeroHijos=2;
$sueldo=4500.10;
$estado=true;
```

Se han declarado las variables para el nombre del trabajador, fecha de nacimiento, número de hijos y el sueldo de un empleado. Tenga en cuenta que las variables siempre inician con el símbolo **$** y no se declaran, también debe tener cuidado con los valores de tipo cadena, cuyos valores deben estar encerrados entre comillas simples o dobles.

En caso de los números enteros y fraccionarios, no deben estar encerrados entre comillas pues serían considerados como cadena de caracteres.

Sólo fines educativos - FreeLibros

```
echo 'Valor de la variables Tipo de Datos';
echo '<br>-----------------------------------------------------';
echo '<br>'.$trabajador.' > '.gettype($trabajador);
echo '<br>'.$fechaNac. ' > '.gettype($fechaNac);
echo '<br>'.$numeroHijos. ' > '.gettype($numeroHijos);
echo '<br>'.$sueldo. ' > '.gettype($sueldo);
echo '<br>'.$estado. ' > '.gettype($estado);
```

La función **echo** permite imprimir un resultado al cliente, aquí también debe considerar que para expresar un valor resultante lo puede realizar por comillas dobles o simples.

La etiqueta **
** se asigna para cada cambio de línea y la función **gettype** devuelve el tipo de datos interpretado por PHP, según el valor asignado a dicha variable.

Veamos otra forma de expresar el mismo código:

```
echo "Valor de la variables Tipo de Datos";
echo "<br>-----------------------------------------------------";
echo "<br>$trabajador >".gettype($trabajador);
echo "<br>$fechaNac >".gettype($fechaNac);
echo "<br>$numeroHijos >".gettype($numeroHijos);
echo "<br>$sueldo >".gettype($sueldo);
echo "<br>$estado >".gettype($estado);
```

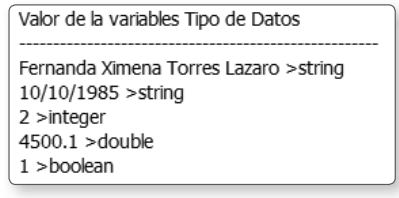
Cuando se usa la doble comilla en la función **echo**, tanto las etiquetas de HTML y las variables de PHP son interpretadas por el compilador de tal forma que pueden convivir dentro de un solo bloque de comillas.

También se implementó un archivo CSS que permitiera formatear el tipo de letra y tamaño de los textos en la impresión:

Archivo: **estilo.css**

```
body{
    font-family: "tahoma";
    font-size: 14px;
}
```

El resultado de la aplicación es:



Sólo fines educativos - FreeLibros

3.8 Manejo de constantes

Una constante es un tipo de variable que mantiene su valor durante la ejecución de una aplicación.

Consideraciones para la declaración de una variable:

- Toda constante de PHP no inicia con el símbolo dólar ($).
- No pueden definirse dos o más veces como las variables.
- No podrán contener elementos de colección.
- A diferencia de las variables, y por convención universal, siempre se escribirán en mayúsculas.

El formato para la declaración de una constante PHP es:

```
define("CONSTANTE",valorConstante);
```

Veamos algunos ejemplos de declaración de constantes:

Declaración	Descripción
define("IVA",0.18)	Se declara la constante IVA con un valor constante de 0.18.
define("PI",3.1416)	Se declara la constante PI con el valor constante de 3.1415.
define("ESTADO","Soltero")	Se declara la contante ESTADO con el valor de cadena **Soltero**.

3.9 Casos desarrollados

Para la implementación de los casos desarrollados usaremos el siguiente estilo, el cual será incorporado en todas las aplicaciones.

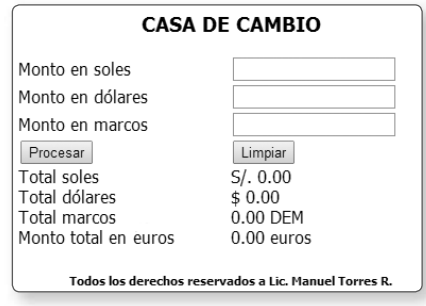
Archivo: **estilo.css**

```
body{
    font-family: "tahoma";
    font-size: 16px;
}
#centrado{
    text-align: center;
}
```

- **Caso desarrollado 1: Diferencia entre echo y printf**

Diseñar una aplicación web con PHP que permita determinar el monto total recaudado en euros, a partir de tres cantidades de dinero ingresados por el usuario, como soles, dólares y marcos.

La interfaz gráfica inicial es la siguiente:



Sólo fines educativos - FreeLibros

Tener en cuenta:

- Debe usar las funciones **echo** y **printf** para la impresión de los resultados.
- Implemente un formulario y una tabla de 2 columnas por 8 filas para mostrar de la mejor manera posible la aplicación.
- Utilice el archivo **estilo.css** para definir los formatos del caso.
- Utilice las siguientes conversiones:

1 dólar = 3.51 soles
1 dólar = 1.09 euros
1 dólar = 2.12 marcos

Archivo: **cambio.php**

```
<!DOCTYPE html>
<html>
    <head>
        <meta charset="UTF-8">
        <link rel="stylesheet" href="estilo.css">
        <title>Diferencia entre echo y printf</title>
    </head>
    <body>
      <header>
        <h3 id="centrado">CASA DE CAMBIO</h3>
      </header>
      <section>
       <form name="frmMontos" method="POST" action="cambio.php">
         <table border="0" cellpadding="0"
                 cellspacing="0" align="center">
           <tr>
            <td width="200">Monto en soles</td>
            <td width="200">
                <input type="text" name="txtSoles"  /></td>
            </tr>
            <tr>
                <td>Monto en dólares</td>
                <td><input type="text" name="txtDolares"  /></td>
            </tr>
            <tr>
                <td>Monto en marcos</td>
                <td><input type="text" name="txtMarcos" /></td>
            </tr>
            <tr>
                <td><input type="submit" value="Procesar" /></td>
                <td><input type="reset" value="Limpiar" /></td>
            </tr>
            <?php
             error_reporting(0);
             $soles=$_POST['txtSoles'];
             $dolares=$_POST['txtDolares'];
             $marcos=$_POST['txtMarcos'];
             $euros=(($soles/3.51)+$dolares+($marcos/2.12))*1.09;
            ?>
            <tr>
              <td>
                Total soles<br>
                Total dólares<br>
```

Sólo fines educativos - FreeLibros

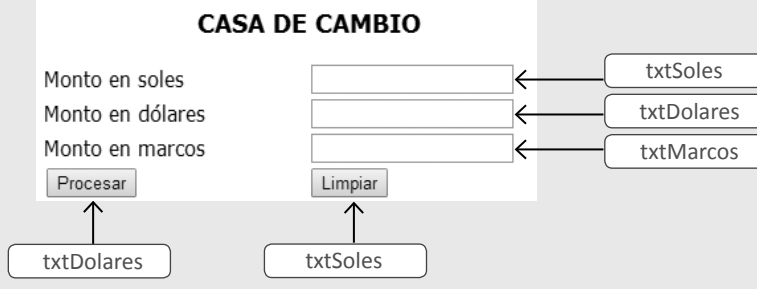
```
                Total marcos<br>
                Monto total en Euros
              </td>
              <td>
               <?php echo "S/. ".number_format($soles, 2)."<br>";
                     echo "$ ".number_format($dolares, 2)."<br>";
                     echo number_format($marcos, 2)." DEM<br>";
                     echo number_format($euros, 2)." euros";
               ?>
              </td>
            </tr>
          </table>
        </form>
      </section>
      <footer>
        <h6 id="centrado">
              Todos los derechos reservados
              a Lic. Manuel Torres R.</h6>
      </footer>
    </body>
</html>
```

Comentarios:

```
error_reporting(0);
$soles=$_POST['txtSoles'];
$dolares=$_POST['txtDolares'];
$marcos=$_POST['txtMarcos'];
$euros=(($soles/3.51)+$dolares+($marcos/2.12))*1.09;
```

La instrucción **error_reporting** permite omitir los mensajes de error ocasionados en una aplicación PHP, esto se debe a que inicialmente la aplicación presentará errores de advertencia ya que las variables no tienen valor, pero sin embargo le hemos mandado a imprimir con la función **echo**.

Luego se capturan los valores registrados en los controles del formulario por medio de la función **$_POST**. Hay que tener en cuenta que cada control del formulario debe tener un nombre asignado, tal como se muestra a continuación:



Sólo fines educativos - FreeLibros

```
echo "S/. ".number_format($soles, 2)."<br>";
echo "$ ".number_format($dolares, 2)."<br>";
echo number_format($marcos, 2)." DEM<br>";
echo number_format($euros, 2)." euros";
```

Usamos la función **echo** para la impresión de los resultados, y por cada línea impresa deberá realizar un cambio de línea con la etiqueta **
**.

La función **number_format** permite preformatear un número de tal manera que se muestre con dos decimales para todo valor monetario.

Si queremos usar el método **printf** para la impresión de los resultados, el código será como sigue:

```
printf("S/. %.2f",$soles);
printf("<br>$ %.2f",$dolares);
printf("<br> %.2f",$marcos);
printf("<br> %.2f",$euros);
```

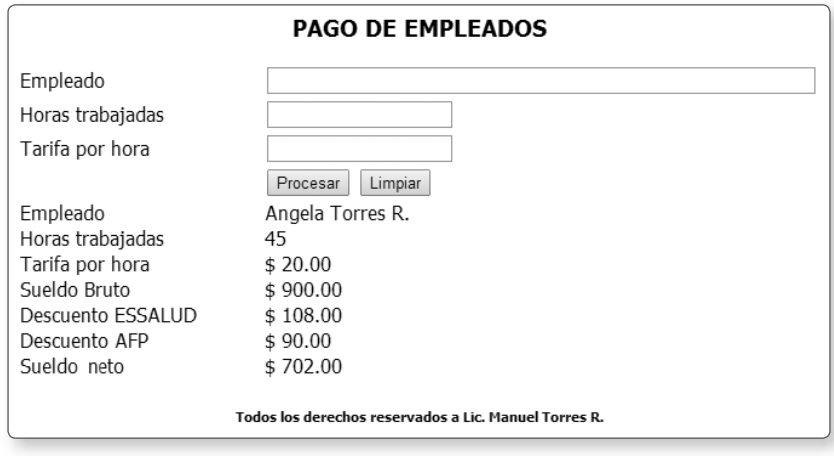
Usamos la función **printf** para la impresión de los resultados de tal forma que el cambio de línea se realiza dentro de la función como un primer parámetro en la cual también se especifica el formato de impresión como:

"%.2F" donde ".2" es para especificar a cuántos decimales imprimirá el resultado y como segundo parámetro el valor a mostrar.

- **Caso desarrollado 2: Manejo de variables y operadores**

Diseñar una aplicación web con PHP que calcule el sueldo bruto, el descuento por ESSALUD, el descuento por AFP y el sueldo neto del empleado de una empresa.

La interfaz gráfica inicial es la siguiente:



Sólo fines educativos - FreeLibros

Tener en cuenta:

- Use variables y operadores en el desarrollo del script en PHP.
- Use formulario y una tabla de 2 columnas por 11 filas para mostrar de la mejor manera posible la aplicación.
- El sueldo bruto se calcula multiplicando el número de horas trabajadas por una tarifa horaria.
- El descuento por ESSALUD es igual al 12 % del sueldo bruto.
- El descuento por AFP es igual al 10 % del sueldo bruto.
- El sueldo neto es la diferencia entre el sueldo bruto y el descuento total.

Archivo: **pago.php**

```
<!DOCTYPE html>
<html>
<head>
   <meta charset="UTF-8">
   <link href="estilo.css" rel="stylesheet">
   <title>Manejo de variables y operadores</title>
</head>
<body>
  <header>
     <h3 id="centrado">PAGO DE EMPLEADOS</h3>
  </header>
  <section>
     <form name="frmPago" method="POST">
        <table border="0" cellpading="0"
                          cellspacing="0" align="center">
         <tr>
           <td width="200">Empleado</td>
           <td><input type="text" name="txtEmpleado"size="70"/></td>
         </tr>
         <tr>
           <td>Horas trabajadas</td>
           <td><input type="text" name="txtHoras" /></td>
         </tr>
         <tr>
           <td>Tarifa por hora</td>
           <td><input type="text" name="txtTarifa" /></td>
         </tr>
         <tr>
           <td></td>
           <td><input type="submit" value="Procesar" />
               <input type="reset" value="Limpiar" />
           </td>
         </tr>
         <?php
            error_reporting(0);
            $empleado=$_POST['txtEmpleado'];
            $horas=$_POST['txtHoras'];
            $tarifa=$_POST['txtTarifa'];

            $sueldoBruto=$horas*$tarifa;
            $descuentoEssalud=$sueldoBruto*0.12;
            $descuentoAFP=$sueldoBruto*0.10;

            $sueldoNeto=$sueldoBruto-$descuentoEssalud-$descuentoAFP;
         ?>
```

Sólo fines educativos - FreeLibros

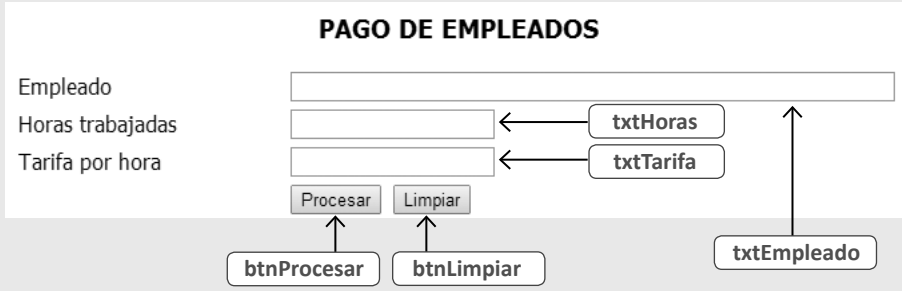
```
            <tr>
               <td>Empleado</td>
               <td><?php echo $empleado; ?></td>
            </tr>
            <tr>
               <td>Horas trabajadas</td>
               <td><?php echo $horas; ?></td>
            </tr>
            <tr>
               <td>Tarifa por hora</td>
               <td><?php echo "$ ".number_format($tarifa,2); ?></td>
            </tr>
            <tr>
              <td>Sueldo Bruto</td>
              <td><?php echo "$ ".number_format($sueldoBruto,2); ?></td>
            </tr>
            <tr>
              <td>Descuento ESSALUD</td>
              <td><?php echo "$ "
                            .number_format($descuentoEssalud,2); ?>
              </td>
            </tr>
            <tr>
              <td>Descuento AFP</td>
              <td><?php echo "$ ".number_format($descuentoAFP,2);?></td>
            </tr>
            <tr>
              <td>Sueldo Neto</td>
              <td><?php echo "$ ".number_format($sueldoNeto,2);?></td>
            </tr>
          </table>
            </form>
      </section>
      <header>
          <h6 id="centrado">Todos los derechos reservados a
                            Lic. Manuel Torres R.</h6>
      </header>
   </body>
  </html>
```

Comentarios:

```
error_reporting(0);
$empleado=$_POST['txtEmpleado'];
$horas=$_POST['txtHoras'];
$tarifa=$_POST['txtTarifa'];

$sueldoBruto=$horas*$tarifa;
$descuentoEssalud=$sueldoBruto*0.12;
$descuentoAFP=$sueldoBruto*0.10;

$sueldoNeto=$sueldoBruto-$descuentoEssalud-$descuentoAFP;
```

Sólo fines educativos - FreeLibros

Empezamos por omitir los errores de advertencia del compilador PHP, luego capturamos los valores registrados en el formulario:



Luego calculamos el sueldo bruto en base a las horas y la tarifa, el descuento de ESSALUD del 12 % en base al sueldo bruto, descuento de AFP del 10 % en base al sueldo bruto y, finalmente, el sueldo neto descontando el descuento de Essalud y AFP al sueldo bruto.

```
<tr>
  <td>Empleado</td>
  <td><?php echo $empleado; ?></td>
</tr>
```

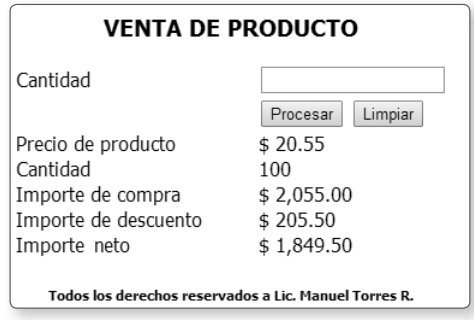
Empleado	Angela Torres R.

Para imprimir los resultados usamos las celdas correspondientes a la tabla, y para imprimir el valor resultante de PHP colocamos la etiqueta **<?php ?>** dentro de la celda **<td>**. No se olvide que para imprimir un valor debe usar la función **echo**.

- **Caso desarrollado 3: Manejo de constantes**

Diseñar una aplicación web con PHP para una tienda que vende un determinado producto, cuyo costo unitario es S/. 20.55. Como oferta, la tienda ofrece un descuento fijo del 10 % del importe de la compra. Se deberá determinar el importe de la compra, el importe del descuento y el importe a pagar por la compra de cierta cantidad de unidades del producto.

La interfaz gráfica inicial es la siguiente:



Sólo fines educativos - FreeLibros

Tener en cuenta:

- Use constante para la definición del costo unitario del producto.
- Use formulario y una tabla de 2 columnas por 7 filas para mostrar de la mejor manera posible la aplicación.
- El importe de compra es el producto de la cantidad más el precio establecido como constante.
- El importe de descuento es el 10 % aplicado al importe de compra.
- El importe neto resulta de sustraer el importe de descuento del importe de compra.

Archivo: **venta.php**

```
<!DOCTYPE html>
<html>
<head>
   <meta charset="UTF-8">
   <title>Manejo de Constantes</title>
   <link href="estilo.css" rel="stylesheet">
</head>
<body>
  <header>
     <h3 id="centrado">VENTA DE PRODUCTO</h3>
  </header>
  <section>
     <form name="frmVenta" method="POST">
        <table border="0" cellpading="0"
              cellspacing="0" align="center">
         <tr>
           <td width="200">Cantidad</td>
           <td><input type="text" name="txtCantidad"  /></td>
         </tr>
         <tr>
           <td></td>
           <td>
                <input type="submit" value="Procesar" />
                <input type="reset" value="Limpiar" />
           </td>
         </tr>
         <?php
           error_reporting(0);
           define("COSTOUNITARIO",20.55);
           define("DESCUENTO",0.10);

           $cantidad=$_POST['txtCantidad'];

           $importeCompra=COSTOUNITARIO*$cantidad;
           $importeDescuento=$importeCompra*DESCUENTO;
           $importePagar=$importeCompra-$importeDescuento;
         ?>
         <tr>
           <td>Precio de producto</td>
           <td><?php echo "$ ".COSTOUNITARIO; ?></td>
         </tr>
         <tr>
           <td>Cantidad</td>
           <td><?php echo $cantidad; ?></td>
         </tr>
         <tr>
```

Sólo fines educativos - FreeLibros

```
          <td>Importe de compra</td>
          <td><?php echo "$ ".number_format($importeCompra,2);?></td>
        </tr>
        <tr>
          <td>Importe de descuento</td>
          <td><?php echo "$ "
                        .number_format($importeDescuento,2); ?>
          </td>
        </tr>
        <tr>
          <td>Importe Neto</td>
          <td><?php echo "$ "
                        .number_format($importePagar,2); ?>
          </td>
        </tr>
      </table>
    </form>
  </section>
  <header>
    <h6 id="centrado">Todos los derechos reservados a
                    Lic. Manuel Torres R.</h6>
  </header>
</body>
</html>
```

Comentarios:

```
error_reporting(0);
define("COSTOUNITARIO",20.55);
define("DESCUENTO",0.10);

$cantidad=$_POST['txtCantidad'];

$importeCompra=COSTOUNITARIO*$cantidad;
$importeDescuento=$importeCompra*DESCUENTO;
$importePagar=$importeCompra-$importeDescuento;
```

Iniciamos omitiendo los errores de advertencia del compilador PHP, luego inicializamos las constantes COSTOUNITARIO y DESCUENTO con los montos indicados en el caso; hay que tener en cuenta que las constantes deben ser declaradas en mayúsculas para diferenciarlas de las variables comunes de la aplicación.

Luego capturamos la cantidad ingresada en el control **txtCantidad** y calculamos el importe de compra multiplicando la constante COSTOUNITARIO por la cantidad ingresada por el usuario; el importe de descuento es el porcentaje del valor constante DESCUENTO, y el importe total a pagar resta ambos valores.

```
<tr>
   <td>Precio de producto</td>
   <td><?php echo "$ ".COSTOUNITARIO; ?></td>
</tr>
```

Para imprimir todos los valores calculados usaremos la tabla, y no olvide que para colocar la impresión con la función **echo** de PHP, se realizará dentro de la etiqueta **<td>** usando la expresión **<?php ?>**.

Sólo fines educativos - FreeLibros

CAP.

4

Estructuras condicionales

Sólo fines educativos - FreeLibros

Sólo fines educativos - FreeLibros

4.1 DEFINICIÓN DE LÓGICA BOOLEANA

Toda aplicación web contiene una condición lógica, por ejemplo, al comparar si el usuario y la clave son correctos, o si el valor registrado es el esperado por el servidor y otras opciones que se presentará; en cualquiera de los casos, el resultado puede ser un true(1) o un false(0). Es decir, solo puede resultar uno por vez.

Veamos la tabla de verdad para la **Y** lógica, donde podremos observar cuándo debe resultar verdadero o falso una condición lógica:

P	Q	P Y Q
V	V	V
V	F	F
F	V	F
F	F	F

Si tenemos A=10, B=5 y C=20 entonces podríamos tener las siguientes condiciones:

CONDICIÓN	DESCRIPCIÓN
(A>B) y (A>C)	El resultado es **falso** por la siguiente razón: Primero, veamos la condición lógica con los valores para poder obtener un resultado: (10>5) y (10>20) V y F Segundo, evaluamos los resultados y lo comparamos con la tabla de verdad: V y F = Falso
(B<A) y (B<C) y (B<0)	El resultado es **falso** por la siguiente razón: Primero, veamos la condición lógica con los valores para poder obtener un resultado: (5<10) y (5<20) y (5<0) V y F y F Segundo, evaluamos los resultados y lo comparamos con la tabla de verdad: V y F y F Ahora observamos que hay tres respuestas con el mismo nivel de prioridad; por lo tanto, empezaremos a evaluar de izquierda a derecha, de la siguiente manera: (V y F) y F V y F Finalmente quedaron V y F, dando como resultado según la tabla de verdad F; por lo tanto, la condición es **falsa**.
(A>0) y (B>0) y (C>0)	El resultado es **verdadero** por la siguiente razón: Primero, veamos la condición lógica con los valores para poder obtener un resultado: (10>0) y (5>0) y (20>0) V y V y V Segundo, si evaluamos que las respuestas nos dan tres verdaderos y seleccionamos la primera pareja, de izquierda a derecha, resolvemos que la condición lógica tiene como resultado V; por lo tanto, la condición es **verdadera.**

Sólo fines educativos - FreeLibros

Ahora veamos la tabla de verdad para la **O** lógica, donde podremos observar cuándo debe resultar verdadera o falsa una condición lógica:

P	Q	P o Q
V	V	V
V	F	V
F	V	V
F	F	F

Si tenemos A=10, B=5 y C=20 entonces podríamos tener las siguientes condiciones:

Condición	Descripción
(A>B) o (A>C)	El resultado es **verdadero** por la siguiente razón: Primero, veamos la condición lógica con los valores para poder obtener un resultado: (10>5) y (10>20) V o F Segundo, evaluamos los resultados y lo comparamos con la tabla de verdad: V o F = Verdadero Eso quiere decir que solo nos bastará tener un resultado verdadero para que la condición lógica sea verdadera.
(B>A) o (B>C) o (B<0)	El resultado es **falso** por la siguiente razón: Primero, veamos la condición lógica con los valores para poder obtener un resultado: (5>10) y (5>20) y (5<0) F y F y F Segundo, evaluamos los resultados y lo comparamos con la tabla de verdad: F y F y F Finalmente, cuando los resultados son F según la tabla de verdad, el resultado es F; por lo tanto, la condición es **falsa**.

Por último, veamos la tabla de verdad para la negación, donde podremos observar cuándo debe resultar verdadera o falsa una condición lógica:

P	Not P
V	F
V	F
F	V
F	V

Sólo fines educativos - FreeLibros

Si tenemos A=10, B=5 y C=20 entonces podríamos tener las siguientes condiciones:

CONDICIÓN	DESCRIPCIÓN
Not (A>B y A>C)	El resultado es **verdadero** por la siguiente razón: Primero, veamos la condición lógica con los valores para poder obtener un resultado: Not ((10>5) y (10>20)) V y F Segundo, evaluamos los resultados: Not (V y F) Not (F) = Verdadero
(Not (A<B)) o A<C	El resultado es **verdadero** por la siguiente razón: Primero, veamos la condición lógica con los valores para poder obtener un resultado: (Not (10<5)) o (10<20) Not (F) o (V) Segundo, evaluamos los resultados: Not (F) o V V o V = Verdadero

4.2 ESTRUCTURAR UNA CONDICIÓN LÓGICA EN PHP

Para estructurar una condición lógica en PHP debemos entender correctamente la condición del problema y conocer correctamente los operadores de relación, como: >, >=, <, <=, ==, etc. El formato es:

(Condicion_Logica)

Debemos considerar que una condición lógica debe asignarse a una estructura que lo interprete, como es el caso de la estructura **if**, **while** y sus variaciones.

Características:

- Una condición lógica simple o compuesta debe estar encerrada entre paréntesis.
- Cuando hay dos o más condiciones, se tiene que unir con un operador lógico && o ||.
- Toda condición lógica, ya sea simple o compuesta, siempre emitirá una sola respuesta lógica; por ningún motivo emitirá dos o más valores.

Sólo fines educativos - FreeLibros

4.2.1 Operadores de comparación

Permiten comparar dos o más valores, emitiendo como resultado un valor booleano.

Operador	Descripción	Ejemplo
==	Permite comparar la igualdad entre dos valores del mismo tipo.	if (a==b)
===	Permite comparar la igualdad en valor y en tipo de datos entre dos valores, es llamado también idéntico.	if(a===b)
>	Permite comparar si un valor es mayor que otro del mismo tipo.	if (a>b)
>=	Permite comparar si un valor es mayor o igual que otro del mismo tipo.	if (a>=b)
<	Permite comparar si un valor es menor que otro del mismo tipo.	if (a<b)
<=	Permite comparar si un valor es menor o igual que otro del mismo tipo.	if (a<=b)
!=	Permite comparar si un valor es diferente.	if (a!=b)

4.2.2 Operadores lógicos

Permiten unir dos o más expresiones booleanas, emitiendo como resultado otro valor booleano.

Operador	Descripción	Ejemplo
&&	**Y** lógica, donde el resultado es verdadero solo si todas las condiciones son verdaderos, caso contrario es falso.	if ($n1==$n2)
||	**O** lógica, donde el resultado es falso solo si todas las condiciones son falsas, caso contrario es verdadero.	if($n1===$n2)
!	Operador de negación, el cual permite negar el resultado de una expresión booleana.	if(!($n1==$n2))

4.2.3 Estructurar bloques de código

Veremos cómo se debe implementar bloques de código, los cuales pueden ser instrucciones que se ejecutarán cuando la condición booleana emita un resultado, que puede ser verdadero o falso.

- Debemos considerar que si solo se realizará una acción cuando la condición es verdadera o falsa, tendremos el siguiente formato:

Acción;

Por ejemplo:

- Calcular el sueldo — $sueldo = $horas * $tarifa;
- Imprimir el sueldo — echo $sueldo;

Sólo fines educativos - FreeLibros

También podríamos encerrarlo entre llaves, normalmente esto se da cuando hay más de dos instrucciones, pero si es aplicado cuando hay una instrucción no genera ningún error, más bien nos aseguramos de que la instrucción es la correcta; como por ejemplo:

- Calcular el sueldo { $sueldo = $horas * $tarifa; }
- Imprimir el sueldo { echo $sueldo; }

- Cuando son varias las acciones que se ejecutarán por resultante de la condición lógica, debemos seguir el siguiente formato obligatoriamente:

```
{Accion1;
Accion2;}
```

- Si necesitamos calcular el sueldo e imprimirlo a la vez.

```
{
    $sueldo=$horas * $tarifa;
    echo $sueldo;
}
```

4.2.4 Control de errores

Los diferentes tipos de errores en PHP pueden ser controlados mediante la función **error_reporting**. Veamos las opciones que presenta:

ERROR_REPORTING	DESCRIPCIÓN
error_reporting(0);	Permite desactivar todas las notificaciones de error generadas desde la aplicación PHP.
error_reporting(E_ERROR | E_WARNING | E_PARSE);	Permite notificar los errores ocasionados al momento de ejecutar la aplicación PHP.
error_reporting(E_ERROR | E_WARNING | E_PARSE | E_NOTICE);	Permite notificar los errores, además de aquellos ocasionados cuando una variable ha sido declarada pero no ha sido inicializada (E_NOTICE).
error_reporting(E_ALL ^ E_NOTICE);	Esta configuración se encuentra implementada originalmente dentro del archivo **php.ini**. Permite notificar todos los errores ocultando aquellos provenientes del **E_NOTICE**.
error_reporting(E_ALL);	Permite notificar todos los errores ocasionados por una aplicación PHP.
error_reporting(-1);	Realiza la misma acción que **error_reporting(E_ALL)** pero con la opción de mantenerse exactamente igual a pesar de las nuevas versiones de PHP.

Sólo fines educativos - FreeLibros

4.3 Estructura condicional If simple

Hemos visto que los resultados de una condición lógica pueden ser «verdadero» o «falso». En ese sentido, la condición If simple trabaja solamente en uno de los lados; es decir, en el resultado verdadero de una condición, haciendo que el lado falso no tenga acción y se pueda seguir con las demás instrucciones del PHP. Veamos el formato:

```
if (Condicion_Logica)
    Acción;
```

Observaciones:

- Debemos considerar que la condición lógica usa los operadores lógicos y relacionales.
- Una condición lógica siempre debe emitir como resultado un solo valor lógico, ya sea true o false.
- Toda condición debe estar encerrada entre paréntesis.
- No será necesario asignar llaves cuando se realiza una sola acción.
- Si se realizan varias acciones se debe encerrar entre llaves obligatoriamente.

Las variaciones de la condicional **If** simple pueden ser:

if (Condicion_Logica){
Acciones;
}
	Aquí se podrán definir dos o más acciones cuando la condición lógica resulta verdadera.

if (Condicion_Logica1 && Condicion_Logica2) Acción;	Aquí la condición lógica está asociada a otra, de tal manera que siempre emita un resultado, tanto verdadero o falso, ejecutando al final una sola acción. En algunas ocasiones los desarrolladores pueden usar esta misma expresión condicional en una sola línea.

if (Condicion_Logica1 || Condicion_Logica2){
Acciones;
}
	Aquí la condición lógica se asocia a otra de forma que ejecuta las acciones implementadas, siempre y cuando cumpla con la condición lógica; nótese que deben estar encerradas entre llaves las acciones.

Veamos algunos ejemplos:

- Se necesita condicionar la edad de una persona mostrando el mensaje «Mayor de edad» solo si la edad es mayor o igual a 18.

```
$edad=21;
if ($edad>17) echo "Mayor de edad";
```

- Se necesita condicionar la evaluación de un determinado alumno, si se encuentra desaprobado se le aumentara el residuo de 5 de dicha nota, imprimir el resultado. Un alumno se encuentra desaprobado solo si su evaluación es inferior o igual a 12.

```
$evaluación=10;
if ($evaluación <=12){
     $evaluación = $evaluación + $evaluacion%5;
     echo $evaluación;
}
```

Sólo fines educativos - FreeLibros

- Se necesita validar que la caja de texto donde se ingresará el nombre del usuario no se encuentre vacía; si ese es el caso, emitir el mensaje: «Debe llenar el nombre del usuario».

```
if (empty($_POST['txtNombre'])) {
        $mensaje = "Debe llenar el nombre del usuario";
}
```

- Se necesita validar que el contenido de una caja de texto sea exclusivamente numérico, ya sea entero o decimal; en caso no sea así, emitir el mensaje: «Solo debe ingresar valores numéricos..!!».

```
if(!is_numeric($_POST['txtNumero'])) {
        echo 'Solo debe ingresar valores numéricos..!!';
}
```

- Se necesita determinar si una caja de texto está vacía; en caso fuera así, emitir el siguiente mensaje: «Ingrese un valor..!!».

```
if(!isset($_POST['txtValor'])) {
        echo 'Ingrese un valor..!!';
}
```

- Se necesita asignar un costo, por hora, a un determinado empleado según la categoría del mismo; siendo operario el costo es de $10, administrativo $25 y jefe $50.

```
//Obtener la categoría desde el control menu de selección del HTML5
$categoría=$_POST[menuCategoria];

//Verificar el tipo de categoría
if(categoría=='Operario')  $costo=10;
if(categoría=='Administrativo')  $costo=25;
if(categoría=='Jefe')  $costo=50;
```

4.4 ESTRUCTURA CONDICIONAL IF DOBLE

Como habíamos visto anteriormente, una estructura If tiene dos alternativas según su condición lógica; se dice condicional If doble cuando toma en referencia a ambos valores, claro está que se tomará una u otra opción, pero no las dos a la vez. Veamos el formato:

```
if (Condicion_Logica)
    Accion_Verdadera;
else
    Accion_Falsa;
```

Observaciones:

- Debemos considerar que la condición lógica usa los operadores lógicos y relacionales.
- Las acciones verdaderas solo se ejecutarán cuando la condición lógica tenga como resultado el valor *true*.
- Las acciones falsas se ejecutan cuando el resultado emitido por la condición lógica devuelva *false*.

Sólo fines educativos - FreeLibros

- Considere que las llaves serán necesarias solo si las acciones, en ambos casos, superan a dos.

Las variaciones de la condicional If doble pueden ser:

if (Condicion_Logica){
Acciones_Verdaderas;
}else{
Acciones_Falsas;
}
	La condición lógica se evalúa si resulta ser verdadera; entonces, se ejecutarán las acciones verdaderas siempre encerradas por llaves, ya que son más de dos acciones, caso contrario se ejecutarán las acciones falsas.

if (Condicion_Logica1 && Condicion_Logica2)
Accion_Verdadera;
else
Accion_Falsa;
	La condición lógica se asocia a otra en caso resulte verdadero el resultado, entonces se ejecutará la acción verdadera; no necesita llaves por ser una sola acción, el mismo caso sucede en la acción falsa.
	
	En algunas ocasiones los desarrolladores pueden usar esta misma expresión condicional en una sola línea.

if (Condicion_Logica1 || Condicion_Logica2){
Acciones_Verdaderas;
}else{
Acciones_Falsas;
}
	La condición lógica se asocia a otra; cuando la respuesta sea verdadera se ejecutarán las acciones verdaderas siempre encerradas entre llaves por ser más de dos acciones, el mismo caso sucede cuando el resultado sea falso.

Veamos algunos ejemplos:

- Se necesita condicionar que una persona tenga como año de nacimiento 1980 y a la vez se encuentre casado; mostrar un mensaje de «Nacido en el 80 con estado civil casado»; caso contrario, mostrar el siguiente mensaje: «No cumple con las especificaciones condicionales..!!».

```
$estadoCivil='casado';
$fechaNac ='1980-11-20';
$aFecha = explode("-", $fechaNac);
$año = $aFecha[0];

if ($año==1980 && $estadoCivil=='casado')
    echo 'Nacido en el 80 con estado civil casado';
    else
    echo 'No cumple con las especificaciones condicionales..!!';
```

- Se necesita condicionar la evaluación obtenida por un alumno en un determinado curso; si la nota es inferior o igual a 12 determinar cuántos puntos le falta para aprobar, e imprimir el resultado; caso contrario, determinar por cuántos puntos aprobó la evaluación y mostrar el mensaje «aprobado» además de los puntos solicitados.

```
$evaluación=14;
if ($evaluación<=12){
    $puntos = 13 - $evaluación;
    echo 'Le falta ' .$puntos. ' para aprobar';
}else{
    $puntos = $evaluación - 13;
    echo 'Aprobó por ' .$puntos . ' puntos';
}
```

Sólo fines educativos - FreeLibros

- Se necesita condicionar el sueldo de un empleado, el cual se somete a un descuento del 10 % si su salario sobrepasa los $1800.00; caso contrario, no hay descuento.

```
$sueldo=1900;
if ($sueldo>1800)
     $descuento = $sueldo * 10.0/100.0;
else
     $descuento = 0;
echo 'El descuento es: '.$descuento;
```

4.5 ESTRUCTURA CONDICIONAL IF DOBLEMENTE ENLAZADA

La estructura condicional If doblemente enlazada normalmente se usa cuando la condición lógica tiene muchas alternativas, sea el caso de cargos de los empleados en una empresa (Jefe, Programador, Analista, etc.) o tipos de clientes en un tienda comercial (Corporativo, Eventual, VIP, etc.). La evaluación condicional se realiza de forma *top-down*; es decir, en forma descendente. Se buscará la condición correcta hasta el final; en caso de no encontrar una condición verdadera, se ejecutarán por defecto las instrucciones previamente definidas por el desarrollador.

Formato:

```
if (Condicion_Logica1)
     Accion_Verdadera1;
Elseif (Condicion_Logica2)
    Accion_Verdadera2;
Elseif (Condicion_Logica3)
    Accion_Verdadera3;
Else
Accion_Falsa;
```

Formato gráfico:

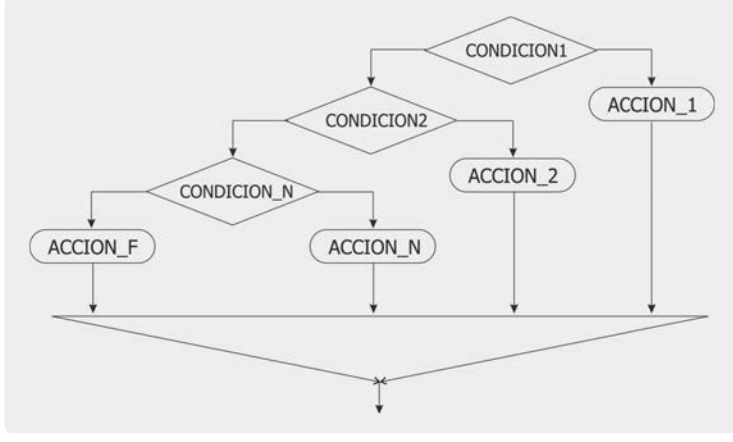


Fig. 4.1 Condicional If doblemente enlazada

Sólo fines educativos - FreeLibros

Observaciones:

- Debemos considerar que todas las condiciones lógicas usan los operadores lógicos y relaciones.
- Cuando una condición lógica resulta *true* se ejecutan las acciones previstas y se termina la evaluación.
- La característica principal de la condicional doblemente enlazada es que cuando no cumple con la primera condicional salta a la siguiente, en descendencia.
- Si no cumple con ninguna de las condicionales se ejecutarán las acciones programadas en el atributo **else**; es decir, se ejecutarán las acciones por defecto.
- Las condiciones lógicas pueden usar una misma variable en todas las comparaciones, pero no deben tener la misma evaluación.

Las variaciones de la condicional If doblemente enlazada pueden ser:

`if (Condicion_Logica1){`
`  Acciones_Verdaderas1;`
`} elseif (Condicion_Logica2) {`
`  Acciones_Verdaderas2;`
`} elseif (Condicion_Logica3) {`
`  Acciones_Verdaderas3;`
`} else {`
`  Acciones_Falsas;`
`}`
	La estructura condicional If doblemente enlazada también controla bloques de instrucciones PHP, siempre que se encuentren encerradas entre llaves. Las acciones falsas solo se ejecutarán cuando ninguna de las condiciones cumple con la lógica del problema, es también conocido como acciones por defecto.

`if (Condicion_Logica1){`
`  Acciones_Verdaderas1;`
`} elseif (Condicion_Logica2) {`
`  Acciones_Verdaderas2;`
`} elseif (Condicion_Logica3) {`
`  Acciones_Verdaderas3;`
`}`
	No necesariamente debe tener acciones por defecto, ya que eso dependerá del desarrollador; esta variación es correcta. En caso no se cumpla con ninguna de la condiciones, se ejecutarán las acciones implementadas seguidamente, después de la estructura condicional doblemente enlazada.

Veamos algunos ejemplos:

- Se necesita asignar un costo, por hora, a un determinado empleado según la categoría del mismo, siendo que: operario $ 10, administrativo $ 25 y jefe $ 50.

```
//Obtener la categoría desde el control menú de selección del HTML5
$categoría=$_POST[menuCategoria];

//Verificar el tipo de categoría
if($categoría=='Operario')
$costo=10;
elseif ($categoría=='Administrativo')
$costo=25;
elseif($categoría=='Jefe')
$costo=50;
```

Sólo fines educativos - FreeLibros

- Se necesita mostrar una fecha en letras con el siguiente formato: «05 de octubre del 2015» cuando la fecha capturada es 2015-10-05.

```
//Capturamos la fecha
$fechaPago='2015-10-05';

//Creamos una partición de la fecha
$mFecha = explode("-", $fechaPago);
$dia = $mFecha[0];
$mes = $mFecha[1];
$año = $mFecha[2];

if ($mes==1)
    $mesLetras='Enero';
   elseif ($mes==2)
            $mesLetras='Febrero';
   elseif ($mes==3)
            $mesLetras='Marzo';
   elseif ($mes==4)
            $mesLetras='Abril';
   elseif ($mes==5)
            $mesLetras='Mayo';
   elseif ($mes==6)
            $mesLetras='Junio';
   elseif ($mes==7)
            $mesLetras='Julio';
   elseif ($mes==8)
            $mesLetras='Agosto';
   elseif ($mes==9)
            $mesLetras='Setiembre';
   elseif ($mes==10)
            $mesLetras='Octubre';
   elseif ($mes==11)
            $mesLetras='Noviembre';
   elseif ($mes==12)
            $mesLetras='Diciembre';

//Imprimir la fecha
echo $dia .' de '. $mesLetras . ' del '.$año;
```

4.6 ESTRUCTURA CONDICIÓN SWITCH

Es una estructura de condición múltiple, usada como alternativa a la estructura If doblemente enlazada. Su formato es:

```
switch ($Variable a evaluar){
 case valor1: accion1; break;
 case valor2: accion2; break;
 case valor…: accion…; break;
 case valorN: accionN; break;
 default: accion_Falsa;
}
```

Sólo fines educativos - FreeLibros

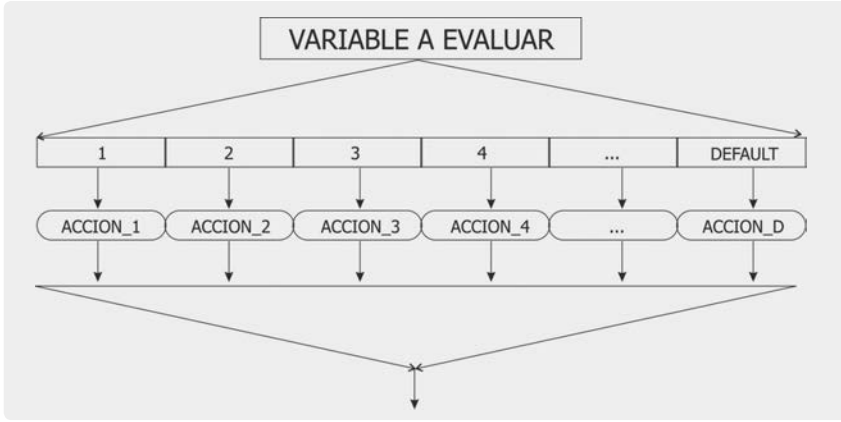


Fig. 4.1 Sentencia múltiple switch

Observaciones:

- Switch busca dentro de sus opciones un valor parecido a la variable a evaluar, una vez encontrado se termina el ciclo de evaluación y se ejecutan las acciones previstas en el caso.
- Cada caso debe tener la sentencia **break** para indicar que se terminan las acciones previstas.
- La no asignación de la sentencia **break** en los casos permitirá que siga buscando la igualdad; por lo tanto, se ejecutarán todas las acciones programadas hasta encontrar el primer **break**.
- También es posible usar la sentencia **continue**, pero se recomienda su uso cuando se implemente el switch dentro de un ciclo de repeticiones como **for** o **while**, ya que puede hacer un salto dentro del ciclo, a diferencia de la sentencia **break**, la cual permitiría terminar el ciclo de repeticiones.
- No es necesario asignar un **break** al último caso, así sea un caso de opción por defecto.

Las variaciones de la condicional switch pueden ser:

```
switch ($Variable a evaluar){
 case valor1:
    accion1.1;
    accion1.2;
    accion1.3;
    break;
 case valor2:
    accion2.1;
    accion2.2;
    accion2.3;
    break;
default:
    accion_Falsa1;
    accion_Falsa2;
}
```

Cuando se especifican dos o más acciones por cada caso, no será necesario encerrarlos entre llaves, pues las acciones se ejecutarán hasta encontrar un **break**.

Lo que sí debemos considerar es que si dentro de un caso se implementa una condicional, se deberá respetar las reglas impuestas por dicha condicional.

```
switch ($Variable a evaluar){
 case valor1: accion1; break;
 case valor2: accion2; break;
 case valor…: accion3; break;
 case valorN: accionN;
}
```

Si decide no usar el valor por defecto, se evaluarán todos los casos, hasta encontrar la variable; en caso no se encontrase, se seguirá con las demás instrucciones después del switch. Hay que tener en cuenta que el último caso no necesitará la sentencia **break**.

Sólo fines educativos - FreeLibros

`switch ($Variable a evaluar){`
`case valor1: return accion1;`
`case valor2: return accion2;`
`case valor…: return accion…;`
`case valorN: return accionN;`
`default: return accion_falsa;`
`}`
	Cuando los casos presentan la sentencia **return**, ya no será necesario usar los **breaks** porque la sentencia **return** devuelve directamente un valor saliendo de la evaluación.

`switch ($Variable a evaluar){`
`case valor1:`
`case valor2:`
`case valor3: accion; break;`
`}`
	Si en un caso no se especifica la sentencia **break**, entonces debemos entender que seguirá la evaluación hasta encontrar una; por ejemplo, en el caso visto se ejecutará la acción siempre y cuando la variable a evaluar sea igual al **valor1** o al **valor2** o al **valor3**.

Veamos algunos ejemplos:

- Se necesita asignar un costo, por hora, a un determinado empleado según la categoría del mismo, siendo que: operario $10, administrativo $25 y jefe $50.

```
//Obtener la categoría desde el control menú de selección del HTML5
$categoría=$_POST[menuCategoria];

//Verificar el tipo de categoría
switch($categoria){
  case 'Operario'            : $costo=10; break;
  case 'Administrativo'      : $costo=25; break;
  case 'Jefe'                : $costo=50;
}
```

O también se podría implementar de la siguiente manera:

```
//Obtener la categoría desde el control menú de selección del HTML5
$categoría=$_POST[menuCategoria];

//Verificar el tipo de categoría
switch($categoria){
  case 'Operario'            : $costo=10; break;
  case 'Administrativo'      : $costo=25; break;
  default                    : $costo=50;
}
```

Note en este último código que no se hace referencia al caso **jefe**, pues según el problema son solo tres casos; así es que el último se toma como valor por defecto.

Sólo fines educativos - FreeLibros

- Se necesita mostrar una fecha en letras con el siguiente formato «05 de Octubre del 2015» cuando la fecha capturada es 2015-10-05.

```
//Capturamos la fecha
$fechaPago='2015-10-05';

//Creamos una partición de la fecha
$mFecha = explode("-", $fechaPago);
$dia = $mFecha[0];
$mes = $mFecha[1];
$año = $mFecha[2];

switch($mes){
     case 1: $mesLetras='Enero'; break;
     case 2: $mesLetras='Febrero'; break;
     case 3: $mesLetras='Marzo'; break;
     case 4: $mesLetras='Abril'; break;
     case 5: $mesLetras='Mayo'; break;
     case 6: $mesLetras='Junio'; break;
     case 7: $mesLetras='Julio'; break;
     case 8: $mesLetras='Agosto'; break;
     case 9: $mesLetras='Setiembre'; break;
     case 10: $mesLetras='Octubre'; break;
     case 11: $mesLetras='Noviembre'; break;
     case 12: $mesLetras='Diciembre';
}

//Imprimir la fecha
echo $dia .' de '. $mesLetras . ' del '.$año;
```

- Se necesita determinar si un número entero menor o igual a 10 es par o impar, mostrando un mensaje para cada caso.

```
//Capturamos el número ingresado por el usuario desde un cuadro de texto
$n=$_POST['txtN'];

//Creamos una partición de la fecha
switch($n){
      case 1:
      case 3:
      case 5:
      case 7:
      case 9: $mensaje='numero es impar'; break;
      case 2:
      case 4:
      case 6:
      case 8:
      case 10: $mensaje='número es par'; break;
      default: $mensaje='número no se encuentre en el rango';
}
```

Sólo fines educativos - FreeLibros

4.7 CASOS DESARROLLADOS

Para la implementación de los casos desarrollados usaremos el siguiente estilo, el cual será incorporado en todas las aplicaciones.

Archivo: **estilo.css**

```
body{
  font-family: tahoma;
  font-size: 14px;
}
#centrado{
  text-align: center;
}
table {
  margin: auto;
}
img{
    margin: auto;
  display: block;
}
```

Caso desarrollado 1: Salario de empleados usando condicional simple

Diseñar una aplicación web con PHP que permita determinar el salario bruto, descuento y salario neto de un empleado, el cual ingresa el número total de horas trabajadas y selecciona la categoría del empleado. Esta categoría determina el costo por hora del empleado.

La interfaz gráfica inicial es la siguiente:

Tener en cuenta:

- El pago por hora trabajada se realizará de acuerdo a la categoría seleccionada para el trabajador.

CATEGORÍA	PAGO POR HORA
Jefe	$ 50.00
Administrativo	$ 30.00
Operario	$ 15.00
Practicante	$ 5.00

Sólo fines educativos - FreeLibros

- El salario bruto se determina mediante el producto de las horas trabajadas; el pago por hora según la categoría del empleado.
- El descuento se calcula en base al 15 % del salario bruto.
- El salario neto es la sustracción del descuento sobre el salario bruto.
- La aplicación no deberá mostrar mensajes de error.
- Al presionar el botón **Calcular,** deberá mostrar los resultados correctos y, a la vez, los valores ingresados en el formulario deberán mantenerse después de mostrado el resultado de la aplicación.
- Los valores monetarios en la aplicación deberán mostrarse en el formato adecuado, es decir, $ 1582.69 en vez de 1582.69.
- La aplicación debe mostrar los resultados tal como se muestra en la siguiente imagen:

Archivo: **salario.php**

```
<!DOCTYPE html>
<html>
<head>
  <meta charset="UTF-8">
  <title>Pago de Empleados</title>
  <link href="estilo.css" rel="stylesheet">
</head>
<body>
  <header>
     <h2 id="centrado">PAGO DE SALARIO DE EMPLEADOS</h2>
     <img src="empleados.jpg" width="600" height="150" />
  </header>
  <section>
     <?php
        error_reporting(0);

        $empleado=$_POST['txtEmpleado'];
        $categoria=$_POST['selCategoria'];
        $horas=$_POST['txtHoras'];

        if ($categoria=='Jefe')
           $selJ='SELECTED';
```

Sólo fines educativos - FreeLibros

```
            else
            $selJ="";
         if ($categoria=='Administrativo')
            $selA='SELECTED';
            else
            $selA="";
         if ($categoria=='Operario')
            $selO='SELECTED';
            else
            $selO="";
         if ($categoria=='Practicante')
            $selP='SELECTED';
            else
            $selP="";
      ?>
      <form method="POST" name="frmSalario" action="salario.php">
        <table border="0" cellspacing="0" cellpadding="0" >
          <tr>
            <td width="150">Empleado</td>
            <td><input type="text" name="txtEmpleado" size="70"
                       value="<?php echo $empleado; ?>" />
            </td>
          </tr>
          <tr>
            <td>Horas</td>
            <td><input type="text" name="txtHoras"
                       value="<?php echo $horas; ?>" /></td>
          </tr>
          <tr>
            <td>Categoria</td>
            <td>
            <select name="selCategoria">
            <option value="Jefe" <?php echo $selJ;?> >Jefe</option>
            <option value="Administrativo" <?php echo $selA; ?> >
                       Administrativo </option>
            <option value="Operario" <?php echo $selO; ?> >
                       Operario</option>
            <option value="Practicante" <?php echo $selP; ?> >
                       Practicante</option>
            </select>
            </td>
          </tr>
          <tr>
            <td></td>
            <td>
                 <input type="submit" value="Calcular"
                                      name="btnCalcular"/>
                 <input type="reset" value="Limpiar controles"
                        name="btnLimpiar"/>
            </td>
          </tr>
          <?php
            if ($categoria=='Jefe') $pagoHora=50;
            if ($categoria=='Administrativo') $pagoHora=30;
            if ($categoria=='Operario') $pagoHora=15;
            if ($categoria=='Practicante') $pagoHora=5;

            $sBruto=$pagoHora*$horas;
            $descuento=$sBruto*15.0/100.0;
            $sNeto=$sBruto-$descuento;
          ?>
```

Sólo fines educativos - FreeLibros

```
            <tr>
              <td>Salario Bruto</td>
              <td>
                 <?php echo "$ ".number_format($sBruto, 2,'.','');?>
              </td>
            </tr>
            <tr>
              <td>Descuento</td>
              <td>
               <?php echo "$ ".number_format($descuento,2,'.',''); ?>
              </td>
            </tr>
            <tr>
              <td>Salario Neto</td>
              <td>
               <?php echo "$ ".number_format($sNeto,2,'.','');?></td>
            </tr>
         </table>
      </form>
   </section>
   <footer>
      <h6 id="centrado">Todos los derechos reservados –
                        Lic. Manuel Torres</h6>
   </footer>
  </body>
  </html>
```

Comentarios:

```
error_reporting(0);

$empleado=$_POST['txtEmpleado'];
$categoria=$_POST['selCategoria'];
$horas=$_POST['txtHoras'];
```

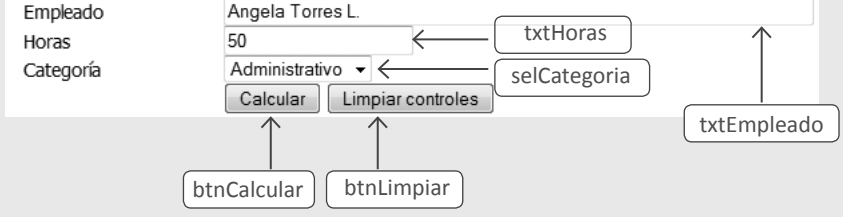
Omitiremos los errores ocasionados por la impresión de variables antes de ser llenadas de valor, si observamos la siguiente imagen notamos que el error se ocasiona por la no definición de la variable **txtEmpleado**.

Notice: Undefined index: txtEmpleado in C:\wamp\www\Capitulo04\salario.php on line 19

Call Stack

#	Time	Memory	Function	Location
1	0.0006	258128	{main}( )	..\salario.php:0

Para evitar este error usamos la instrucción **error_reporting(0)**. Luego capturamos los valores ingresados por el usuario en los controles del formulario usando la función **$_POST**.

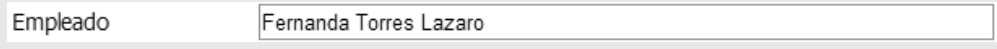


Finalmente, capturamos los valores de los controles usando la función **$_POST** como el nombre del empleado, categoría del empleado y las horas trabajadas.

Sólo fines educativos - FreeLibros

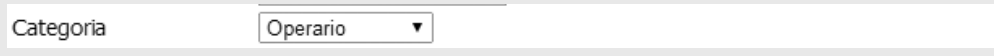
```
if ($categoria=='Jefe')
        $selJ='SELECTED';
        else
        $selJ="";
if ($categoria=='Administrativo')
        $selA='SELECTED';
        else
        $selA="";
if ($categoria=='Operario')
        $selO='SELECTED';
        else
        $selO="";
if ($categoria=='Practicante')
        $selP='SELECTED';
        else
        $selP="";
```

Para poder mantener la categoría seleccionada por el usuario, se debe aplicar la propiedad SELECTED, para lo cual se han creado variables para cada opción de la categoría, por ejemplo, en la categoría **Jefe** la variable sería **$selJ**. Entonces, si el usuario selecciona la categoría **Jefe** se llena la variable **$selJ** del valor SELECTED y así podrá mantenerla seleccionada después de mostrado el resultado.




```
<tr>
          <td width="150">Empleado</td>
          <td><input type="text" name="txtEmpleado" size="70"
                        value="<?php echo $empleado; ?>" />
          </td>
</tr>
```

Aquí se imprime el nombre seleccionado por el usuario en el mismo control **txtEmpleado**, aun después de mostrada la información resultante.




```
<tr>
   <td>Categoria</td>
   <td>
       <select name="selCategoria">
           <option value="Jefe" <?php echo $selJ;?> >Jefe</option>
           <option value="Administrativo" <?php echo $selA; ?> >
                       Administrativo </option>
           <option value="Operario" <?php echo $selO; ?> >
                       Operario</option>
           <option value="Practicante" <?php echo $selP; ?> >
                       Practicante</option>
       </select>
   </td>
</tr>
```

En líneas anteriores usamos variables para asignar la etiqueta SELECTED, según la opción del menú de categorías seleccionadas; ahora se le asignará a cada una de las opciones después del atributo VALUE, ya que el usuario podría seleccionar cualquiera de ellas.

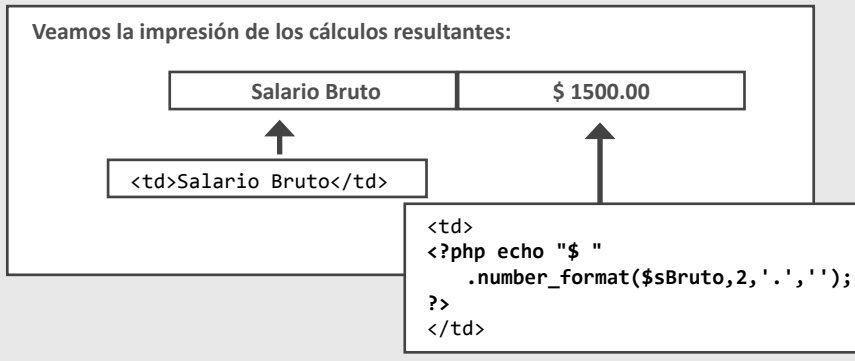
Sólo fines educativos - FreeLibros

```
if ($categoría=='Jefe') $pagoHora=50;
if ($categoría=='Administrativo') $pagoHora=30;
if ($categoría=='Operario') $pagoHora=15;
if ($categoría=='Practicante') $pagoHora=5;

$sBruto=$pagoHora*$horas;
$descuento=$sBruto*15.0/100.0;
$sNeto=$sBruto-$descuento;
```

Ahora asignaremos el pago, por hora, según la categoría seleccionada. Como observamos, se ha evaluado una a una las categorías, ya que estamos evaluando la condicional simple del If, ya que de otra manera podríamos usar condicionales múltiples.

Luego se calcula en salario bruto ($sBruto), descuento ($descuento); recuerde que este se divide con 15.0/100.0 para asegurar un valor fraccionario y finalmente el salario neto ($sNeto).

Veamos la impresión de los cálculos resultantes:



La impresión del valor **salario bruto** pudo haber sido mucho más simple como, por ejemplo:

<?php echo $sBruto; ?> El resultado sería: 1500

La función **number_format** permite controlar el número de decimales y el símbolo de millones de un valor monetario, su formato es:

number_format($variable,Cantidad_decimales,'simbolo_decimal','simbolo_millones');

Finalmente la impresión con **number_format** hace que el número 1500 tome forma de moneda y se imprima correctamente; no olvidarse que la función solo permite asignar decimales pero no muestra el símbolo de la moneda, por eso es que se imprime el símbolo $ directamente, mediante la función **echo**, de la siguiente manera:

<?php echo "$ ".number_format($sBruto,2,'.',''); ?>

- **Caso desarrollado 2: Obsequio a clientes usando condicional simple**

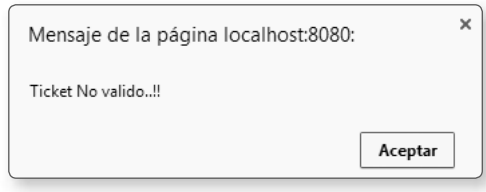
El jefe de *marketing* de una tienda comercial ha propuesto a la gerencia la idea de incentivar a sus clientes con un obsequio, esto con el fin de elevar el número de ventas en el mes de julio, que estadísticamente presenta bajas. Para ello, el cliente al finalizar su compra deberá extraer un *ticket* numerado del 1 al 20 desde una ánfora; en base al número obtenido se determinará el obsequio que recibirá el cliente.

Sólo fines educativos - FreeLibros

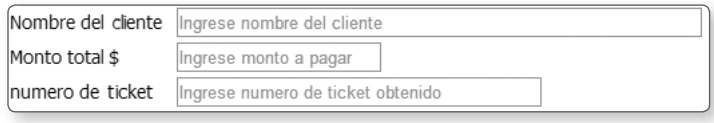
La interfaz gráfica inicial es la siguiente:

Tener en cuenta:

- La aplicación no deberá mostrar mensajes de error.
- Al presionar el botón **Procesar**, deberá mostrar los resultados correctos y, a la vez, los datos ingresados en el formulario deben mantenerse.
- Los valores monetarios en la aplicación deberán mostrarse en el formato adecuado; es decir, $ 1260.42 en vez de 1260.4254.
- Mostrar el mensaje «Ticket no válido» en caso se ingrese un número fuera del rango establecido.



- Deberá mostrar un mensaje indicativo en cada control del formulario, tal como se muestra en la siguiente imagen:



Sólo fines educativos - FreeLibros

- El obsequio asignado según el número de *ticket* es:

Numero de ticket	Descripción del obsequio	Porcentaje de descuento
20	Saco de arroz de 50 kg.	10 %
15 - 19	Caja de leche de 24 latas grandes.	12 %
10 - 14	Aceite 5 litros.	6 %
5 - 9	Saco de azúcar de 50 kg.	13%
1 - 4	Canasta con productos diversos.	16 %

- Finalmente, los resultados deben mostrarse de la siguiente forma:

Archivo: **obsequio.php**

```
<!DOCTYPE html>
<html>
    <head>
        <meta charset="UTF-8">
        <link href="estilo.css" rel="stylesheet">
        <title>Obsequio a Clientes</title>
    </head>
    <body>
        <header>
            <h2 id="centrado">Obsequio a Clientes</h2>
            <img src="tienda.JPG" width="550" height="168" alt="tienda"/>
        </header>
        <?php
            error_reporting(0);
            $monto=$_POST['txtMonto'];
            $ticket=$_POST['txtNumero'];
        ?>
        <section>
            <form name="frmObsequio" action="obsequio.php" method="POST">
            <table border="0" width="550" cellspacing="0"
                   cellpadding="0" align="center">
            <tr>
```

Sólo fines educativos - FreeLibros

```
        <td>Nombre del Cliente</td>
        <td><input type="text" name="txtCliente" size="60"
                   value="<?php echo $_POST[txtCliente]; ?>"
                   placeholder="Ingrese nombre del cliente" />
        </td>
    </tr>
    <tr>
        <td>Monto total $</td>
        <td><input type="text" name="txtMonto"
                   value="<?php echo $monto; ?>"
                   placeholder="Ingrese monto a pagar"/>
        </td>
    </tr>
    <tr>
        <td>numero de Ticket</td>
        <td><input type="text" name="txtNumero" size="40"
              value="<?php echo $ticket; ?>"
              placeholder="Ingrese numero de ticket obtenido"/>
        </td>
    </tr>
    <tr>
        <td></td>
        <td><input type="submit" value="Procesar" /></td>
    </tr>
    <?php
        if ($ticket>=1 && $ticket<=4){
            $obsequio='Canasta con productos diversos';
            $descuento=16.0/100.0 * $monto;
        }

        if ($ticket>=5 && $ticket<=9){
            $obsequio='Saco de azúcar de 50kg.';
            $descuento=13.0/100.0 * $monto;
        }

        if ($ticket>=10 && $ticket<=14){
            $obsequio='Aceite 5 litros.';
            $descuento=6.0/100.0 * $monto;
        }

        if ($ticket>=15 && $ticket<=19){
            $obsequio='Caja de leche de 24 latas grandes.';
            $descuento=12.0/100.0 * $monto;
        }

        if ($ticket==20){
            $obsequio='Saco de arroz de 50kg.';
            $descuento=10.0/100.0 * $monto;
        }

        if ($ticket<1 || $ticket>20){
            echo '<script>
                        alert("Ticket No valido..!!");
                  </script>';
        }

        $nuevoMonto=$monto-$descuento;
    ?>
    <tr>
        <td>Monto a cancelar</td>
        <td>
```

Sólo fines educativos - FreeLibros

```
                    <?php echo '$'.number_format($nuevoMonto,2,'.',''); ?>
                </td>
            </tr>
            <tr>
                <td>Obsequio obtenido</td>
                <td>
                    <?php echo $obsequio; ?>
                </td>
            </tr>
            </table>
        </form>
        </section>
        <footer>
            <h6 id="centrado">Todos los derechos reservados –
                 Lic. Manuel Torres</h6>
        </footer>
    </body>
</html>
```

Comentarios:

```
error_reporting(0);
$monto=$_POST['txtMonto'];
$ticket=$_POST['txtNumero'];
```

Empezaremos omitiendo los errores con la sentencia **error_reporting(0)**, luego capturamos los valores ingresados por el usuario en las variables **monto** y **ticket**, este último almacenará el número de *ticket* que el usuario obtuvo en el sorteo; la variable **monto** almacenará el monto total a pagar por el cliente.

```
<form name="frmObsequio" action="obsequio.php" method="POST">
```

La definición del formulario debe darse de la siguiente manera, primero le asignamos un nombre al formulario, esto se podría obviar pero lo usamos para que todos los controles tengan un nombre adecuado, action define la acción que tomará el formulario después de presionar el botón submit del formulario y, finalmente, el método POST para el envío de información.

```
<input type="text" name="txtCliente" size="60"
       value="<?php echo $_POST[txtCliente]; ?>"
       placeholder="Ingrese nombre del cliente" />

<input type="text" name="txtMonto"
             value="<?php echo $monto; ?>"
       placeholder="Ingrese monto a pagar"/>

<input type="text" name="txtNumero" size="40"
             value="<?php echo $ticket; ?>"
       placeholder="Ingrese numero de ticket obtenido"/>
```

Debemos mostrar los valores ingresados por el usuario y mantenerlos aun después de presionar el botón **Procesar**, por esta razón al atributo **VALUE** se le imprime el valor obtenido desde PHP.

El atributo **PLACEHOLDER** permite mostrar un mensaje en el mismo control, haciendo que el usuario ingrese valores adecuados.

Sólo fines educativos - FreeLibros

```
if ($ticket>=1 && $ticket<=4){
     $obsequio='Canasta con productos diversos';
     $descuento=16.0/100.0 * $monto;
}
if ($ticket>=5 && $ticket<=9){
     $obsequio='Saco de azúcar de 50kg.';
     $descuento=13.0/100.0 * $monto;
}
if ($ticket>=10 && $ticket<=14){
     $obsequio='Aceite 5 litros.';
     $descuento=6.0/100.0 * $monto;
}
if ($ticket>=15 && $ticket<=19){
     $obsequio='Caja de leche de 24 latas grandes.';
     $descuento=12.0/100.0 * $monto;
}
if ($ticket==20){
     $obsequio='Saco de arroz de 50kg.';
     $descuento=10.0/100.0 * $monto;
}
```

Determinaremos el obsequio que se le asigna al cliente según el número de *ticket* obtenido, y aprovecharemos en calcular el descuento, ya que la tabla propuesta cruza ambas condiciones; para este caso, seguimos usando la estructura If simple.

```
if ($ticket<1 || $ticket>20){
      echo '<script>alert("Ticket No valido..!!");</script>';
}
$nuevoMonto=$monto-$descuento;
```

Validamos el número del *ticket* registrado en el formulario; es decir, debe ser mayor o igual a 1, y menor o igual a 20:

En caso el número de *ticket* sea valor inferior a uno o superior a veinte, entonces se estará rompiendo la regla de la aplicación; por lo tanto, debe mostrar un ventana emergente sobre el documento web, esto será realizado con **alert** del Javascript.

```
<td>
 <?php echo '$'
   .number_format($nuevoMonto,2,'.','');
 ?>
</td>

<td>
 <?php echo $obsequio; ?>
</td>
```

Se imprimen los resultados con la función **echo** en la tabla, **number_format** permite mostrar el número en formato de dinero. Finalmente, también se imprime la variable $obsequio obtenido en la condicional anterior.

Sólo fines educativos - FreeLibros

Caso desarrollado 3: Venta de productos usando condicional doble

Una tienda comercial vende solo cinco tipos de productos, los cuales cuentan con la siguiente tabla de precios:

Descripción del producto	Precio unitario
Cocina	$ 1200.00
Refrigeradora	$ 2500.00
Televisión	$ 3200.00
Lavadora	$ 1000.00
Radiograbadora	$ 700.00

Se necesita implementar una aplicación web con PHP que permita determinar el subtotal a pagar por cierta cantidad de productos del mismo tipo, el monto de descuento que asciende al 10 % del subtotal solo si este sobrepasa los $10 000.00; caso contrario, solo el 5 %.

La interfaz gráfica inicial es la siguiente:

Tener en cuenta:

- La aplicación no deberá mostrar mensajes de error.
- Los productos deben mostrarse en un menú de selección.
- No deberá ingresar el precio del producto, ya que será determinado por el producto seleccionado.
- Al presionar el botón **Procesar** deberá mostrar los resultados correctos; y a la vez los datos ingresados en el formulario deben mantenerse a pesar del *refresh* del botón.
- Los valores monetarios en la aplicación deberán mostrarse en el formato adecuado, es decir, $ 1260.42 en vez de 1260.4254.
- Al comprar un producto los resultados deben mostrarse como sigue:

Sólo fines educativos - FreeLibros

Archivo: **tienda.php**

```
<!DOCTYPE html>
<html>
    <head>
        <meta charset="UTF-8">
        <link href="estilo.css" rel="stylesheet">
        <title>Venta de Productos</title>
    </head>
    <body>
        <header>
            <h2 id="centrado">VENTA DE PRODUCTOS ELECTRODOMESTICOS</h2>
            <img src="comercial.jpg" width="600" height="150" />
        </header>
        <?php
            error_reporting(0);

            $cliente=$_POST['txtCliente'];
            $producto=$_POST['selProducto'];
            $cantidad=$_POST['txtCantidad'];

            if ($producto=='Cocina')
                 $selC='SELECTED';
                 else
                   $selC="";
            if ($producto=='Refrigeradora')
                 $selR='SELECTED';
                 else
                   $selR="";
            if ($producto=='Televisión')
                 $selT='SELECTED';
                 else
                   $selT="";
            if ($producto=='Lavadora')
                 $selL='SELECTED';
                 else
                   $selL="";
            if ($producto=='Radiograbadora')
```

Sólo fines educativos - FreeLibros

```
            $selRa='SELECTED';
            else
              $selRa="";
    ?>
    <section>
        <br>
        <form name="frmTienda" action="tienda.php" method="POST">
        <table border="0" width="550" cellspacing="0"
               cellpadding="0" >
        <tr>
            <td>Cliente</td>
            <td>
                <input type="text" name="txtCliente" size="60"
                       value="<?php echo $cliente; ?>" />
            </td>
        </tr>
        <tr>
            <td>Producto</td>
            <td><select name="selProducto">
                 <option value="Cocina" <?php echo $selC;?> >
                        Cocina 6 Hormillas
                 </option>
                 <option value="Refrigeradora" <?php echo $selR;?> >
                        Refrigeradora
                 </option>
                 <option value="Television" <?php echo $selT;?> >
                        Televisión 42
                 </option>
                 <option value="Lavadora" <?php echo $selL;?> >
                        Lavadora 10kg.
                 </option>
                 <option value="Radiograbadora" <?php echo $selRa;?> >
                        Radiograbadora USB
                 </option>
                </select>
            </td>
        </tr>
        <tr>
            <td>Cantidad</td>
            <td><input type="text" name="txtCantidad"
                       value="<?php echo $cantidad; ?>" />
            </td>
        </tr>
        <tr>
            <td></td>
            <td><input type="submit" value="Procesar"
                       name="btnProcesar"/>
            </td>
        </tr>
        <?php
            if ($producto=='Cocina') $precio=1200;
            if ($producto=='Refrigeradora') $precio=2500;
            if ($producto=='Televisión') $precio=3200;
            if ($producto=='Lavadora') $precio=1000;
            if ($producto=='Radiograbadora') $precio=700;

            $subtotal=$precio*$cantidad;

            if ($subtotal>10000)
                $descuento=10/100.0 * $subtotal;
                else
```

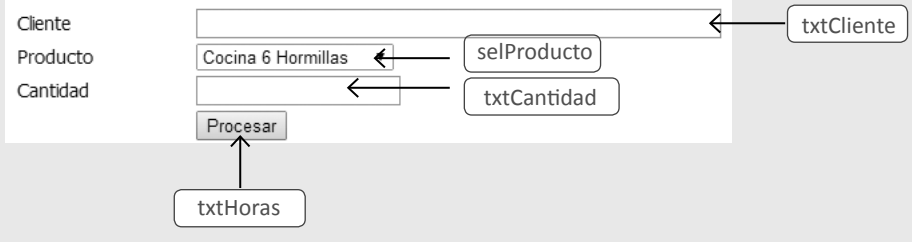
Sólo fines educativos - FreeLibros

```
                    $descuento=5/100.0 * $subtotal;

                $monto=$subtotal-$descuento;
            ?>
            <tr>
              <td>Precio del producto</td>
              <td>
                <?php echo '$'.number_format($precio,2,'.',''); ?>
              </td>
            </tr>
            <tr>
                <td>Subtotal a pagar</td>
                <td>
                  <?php echo '$'.number_format($subtotal,2,'.',''); ?>
                </td>
            </tr>
            <tr>
                <td>Monto de descuento</td>
                <td>
                  <?php echo '$'.  number_format($descuento,2,'.',''); ?>
                </td>
            </tr>
            <tr>
                <td>Monto a pagar</td>
                <td>
                  <?php echo '$'.  number_format($monto,2,'.',''); ?>
                </td>
            </tr>
            </table>
        </form>
        </section>
        <footer>
            <h6 id="centrado">Todos los derechos reservados –
                 Lic. Manuel Torres</h6>
        </footer>
    </body>
</html>
```

Comentarios:

```
error_reporting(0);

$cliente=$_POST['txtCliente'];
$producto=$_POST['selProducto'];
$cantidad=$_POST['txtCantidad'];
```

Omitimos los errores de advertencia con la sentencia **error_reporting(0)**, luego capturamos los valores ingresados por el usuario.



Sólo fines educativos - FreeLibros

```
if ($producto=='Cocina')
        $selC='SELECTED';
        else
        $selC="";
if ($producto=='Refrigeradora')
        $selR='SELECTED';
        else
        $selR="";
if ($producto=='Televisión')
        $selT='SELECTED';
        else
        $selT="";
if ($producto=='Lavadora')
        $selL='SELECTED';
        else
        $selL="";
if ($producto=='Radiograbadora')
        $selRa='SELECTED';
        else
        $selRa="";
```

Asignamos una opción del control **select** de productos de forma predeterminada, para lo cual implementamos el código anterior. Luego debemos mostrar los datos seleccionados en el control **selProducto**, para lo cual implementamos el siguiente código:

```
<select name="selProducto">
 <option value="Cocina" <?php echo $selC;?>>Cocina 6 Hormillas</option>
 <option value="Refrigeradora"<?php echo $selR;?>>Refrigeradora</option>
 <option value="Television" <?php echo $selT;?>>Television 42</option>
 <option value="Lavadora" <?php echo $selL;?>>Lavadora 10kg. </option>
 <option value="Radiograbadora" <?php echo $selRa;?>>Radiograbadora USB <option>
</select>
```

```
if ($producto=='Cocina') $precio=1200;
if ($producto=='Refrigeradora') $precio=2500;
if ($producto=='Televisión') $precio=3200;
if ($producto=='Lavadora') $precio=1000;
if ($producto=='Radiograbadora') $precio=700;

$subtotal=$precio*$cantidad;

if ($subtotal>10000)
        $descuento=10/100.0 * $subtotal;
        else
        $descuento=5/100.0 * $subtotal;

$monto=$subtotal-$descuento;
```

Se asigna el precio de los productos según lo seleccionado por el usuario, aquí todavía usamos la condicional simple, luego calculamos subtotal en base al precio obtenido y la cantidad ingresada por el usuario. El descuento se evalúa con una condicional doble, en la cual si el subtotal supera a 10 000 se le aplica el 10 % al subtotal; caso contrario, solo el 5 %. Finalmente, con todos los valores obtenidos calcularemos el monto total a pagar.

Sólo fines educativos - FreeLibros

```
<td>
 <?php echo '$'
  .number_format($precio,2,'.',''); ?></td>
<td>
 <?php echo '$'
  .number_format($subtotal,2,'.',''); ?></td>
<td>
 <?php echo '$'
  .number_format($descuento,2,'.',''); ?>
</td>
<td>
 <?php echo '$'
  .number_format($monto,2,'.',''); ?> </td>
```

Para la impresión de los resultados se debe usar la tabla implementada en el inicio, en la cual combinaremos código HTML y PHP. Tenga en cuenta que la función **number_format** para definir el formato adecuado del dinero impreso.

- **Caso desarrollado 4: Control de mensualidad usando condicional doblemente enlazada**

Una universidad particular necesita implementar una aplicación web para que los alumnos consulten el monto mensual, el descuento y el monto a cancelar por concepto de pagos mensuales, según la categoría y su promedio ponderado obtenido.

La interfaz gráfica inicial es la siguiente:

Sólo fines educativos - FreeLibros

Tener en cuenta:

- La aplicación no deberá mostrar mensajes de error.
- Implemente un formulario y una tabla de 7 filas por 3 columnas para presentar la aplicación de la mejor manera posible.
- Valide el nombre del alumno mostrando el mensaje «Debe registrar nombre del alumno», cuando este no registre su nombre.
- Valide la categoría del alumno mostrando el mensaje «Debe seleccionar una categoría», cuando este no seleccione su categoría.
- Valide el ingreso del promedio mostrando el mensaje «Debe registrar correctamente el promedio», cuando este no registre adecuadamente el promedio.
- Los costos mensuales según la categoría se muestra en el siguiente cuadro:

Categoría del alumno	Monto mensual
A	$ 850.00
B	$ 750.00
C	$ 650.00
D	$ 500.00

- El porcentaje de descuento se basa en el promedio ponderado obtenido por el alumno, tal como se muestra en el siguiente cuadro:

Promedio	Porcentaje de descuento
Menor a 12	0 %
Entre 13 y 15	10 %
Entre 16 y 17	15 %
Entre 18 y 19	25 %
Igual a 20	50 %

- El resultado se muestra en la siguiente interfaz:

Sólo fines educativos - FreeLibros

Archivo: **universidad.php**

```
<!DOCTYPE html>
<html>
    <head>
        <meta charset="UTF-8">
        <link href="estilo.css" rel="stylesheet">
        <title>Mensualidad de Universidad</title>
    </head>
    <body>
        <header>
            <h2 id="centrado">Mensualidad de Universidad</h2>
            <img src="alumnos.jpg" width="850" height="200" />
        </header>
        <?php
            error_reporting(0);

            $alumno=$_POST['txtAlumno'];
            $categoria=$_POST['selCategoria'];
            $promedio=$_POST['txtPromedio'];

            if ($categoria=='A') $selA='SELECTED'; else $selA="";
            if ($categoria=='B') $selB='SELECTED'; else $selB="";
            if ($categoria=='C') $selC='SELECTED'; else $selC="";
            if ($categoria=='D') $selD='SELECTED'; else $selD="";

            $mAlumno='';
            $mCategoria='';
            $mPromedio='';
            if (empty($alumno))
                $mAlumno='Debe registrar nombre del alumno';
            if($categoría=='seleccione')
                $mCategoria='Debe seleccionar una categoría';
            if(empty($promedio) || !is_numeric($promedio))
                $mPromedio='Debe registrar correctamente el promedio';
                elseif($promedio<0 || $promedio>20)
                $mPromedio='El promedio debe estar entre 0 y 20';
        ?>
        <section>
            <form name="frmUniversidad" method="POST"
                  action="Universidad.php">
            <table border="0" width="800" cellspacing="0"
                   cellpadding="0" aling="center">
            <tr>
                <td width="200">Nombre completo del alumno</td>
                <td width="400"><input type="text"
                                  name="txtAlumno" size="50"
                                  value="<?php echo $alumno; ?>" />
                </td>
                <td  width="200" id="error">
                                <?php echo $mAlumno; ?>
                </td>
            </tr>
            <tr>
                <td>Seleccione categoría</td>
                <td><select name="selCategoria">
                     <option value="seleccione" SELECTED>
                              Seleccione categoria</option>
                     <option value="A" <?php echo $selA;?> >A</option>
                     <option value="B" <?php echo $selB;?> >B</option>
                     <option value="C" <?php echo $selC;?> >C</option>
```

Sólo fines educativos - FreeLibros

```
                    <option value="D" <?php echo $selD;?> >D</option>
                </select></td>
            <td id="error"><?php echo $mCategoria; ?></td>
        </tr>
        <tr>
            <td>Ingrese promedio</td>
            <td><input type="text" name="txtPromedio"
                       value="<?php echo $promedio; ?>" /></td>
            <td id="error"><?php echo $mPromedio; ?></td>
        </tr>
        <tr>
            <td></td>
            <td>
                <input type="submit" value="Procesar"
                       name="btnProcesar"/>
            </td>
        </tr>
        <?php
            if($categoria=='A')
                $monto=850;
            elseif ($categoria=='B')
                $monto=750;
            elseif ($categoria=='C')
                $monto=650;
            elseif ($categoria=='D')
                $monto=500;
            else
                $monto=0;

            if ($promedio<=12)
                $descuento=0;
                elseif ($promedio<=15)
                    $descuento=10.0/100.0 * $monto;
                elseif ($promedio<=17)
                    $descuento=15.0/100.0 * $monto;
                elseif ($promedio<=19)
                    $descuento=25.0/100.0 * $monto;
                elseif ($promedio==20)
                    $descuento=50.0/100.0 * $monto;

            $montoCancelar=$monto-$descuento;
        ?>
        <tr>
            <td>Monto mensualidad</td>
            <td><?php echo '$ '.number_format($monto,2,'.',''); ?></td>
        </tr>
        <tr>
         <td>Monto descuento</td>
         <td><?php echo '$ '.number_format($descuento,2,'.','');?></td>
        </tr>
        <tr>
            <td>Monto a cancelar</td>
            <td>
             <?php echo '$ '.number_format($montoCancelar,2,'.',''); ?>
            </td>
        </tr>
        </table>
    </form>
    </section>
</body>
</html>
```

Sólo fines educativos - FreeLibros

Comentarios:

```
error_reporting(0);

$alumno=$_POST['txtAlumno'];
$categoria=$_POST['selCategoria'];
$promedio=$_POST['txtPromedio'];
```

Omitimos los errores de advertencia con la sentencia **error_reporting(0)**, seguidamente capturamos los valores ingresados por el usuario.

```
if ($categoria=='A') $selA='SELECTED'; else $selA="";
if ($categoria=='B') $selB='SELECTED'; else $selB="";
if ($categoria=='C') $selC='SELECTED'; else $selC="";
if ($categoria=='D') $selD='SELECTED'; else $selD="";
```

Se condiciona la categoría para poder mantener el valor seleccionado por el usuario después de mostrar los resultados.

```
$mAlumno='';
$mCategoria='';
$mPromedio='';

if (empty($alumno)) $mAlumno='Debe registrar nombre del alumno';

if($categoría=='seleccione') $mCategoria='Debe seleccionar una categoría';

if(empty($promedio) || !is_numeric($promedio))
       $mPromedio='Debe registrar correctamente el promedio';
       elseif($promedio<0 || $promedio>20)
       $mPromedio='El promedio debe estar entre 0 y 20';
```

La validación de los controles del formulario se inician declarando las variables para los mensajes $mAlumno(mensaje para el nombre del alumno), $mCategoria(mensaje para la categoría) y $mPromedio(mensaje para el promedio).

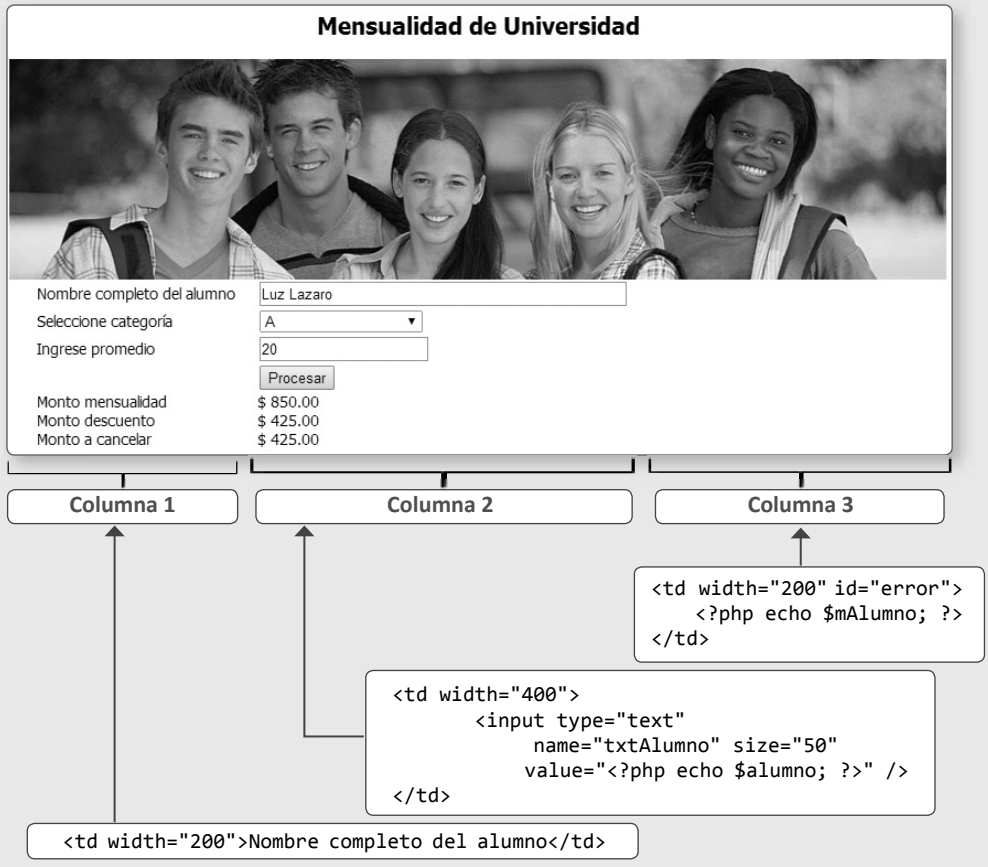
La validación para el nombre del alumno se realiza evaluando el contenido de la variable, para esto usamos la función **empty**, que permite determinar si la variable está vacía o llena. En el caso de la categoría se condiciona sobre la palabra «Seleccione», ya que es la primera opción que presenta las categorías.

Finalmente, para validar el promedio, usamos la combinación de la función **empty** para determinar si el usuario dejó vacío el control y la función **is_numeric**, que permite determinar si el valor a ingresar es un valor numérico o no. En una condición doblemente enlazada, evaluamos si el promedio se encuentra entre 0 y 20, pues de otra manera los valores serían errados.

Sólo fines educativos - FreeLibros

```
<tr>
        <td width="200">Nombre completo del alumno</td>
        <td width="400"><input type="text"
                              name="txtAlumno" size="50"
                              value="<?php echo $alumno; ?>" />
        </td>
        <td  width="200" id="error"><?php echo $mAlumno; ?> </td>
</tr>
```

Al implementar la tabla indicamos que debía tener 3 columnas con la siguiente distribución:



En la tercera columna se imprimirán los mensajes de error ocasionados por no registrar los valores correctos en los controles.

```
<tr>
      <td>Seleccione categoría</td>
      <td><select name="selCategoria">
                  <option value="seleccione" SELECTED>
                        Seleccione categoria</option>
                  <option value="A" <?php echo $selA;?> >A</option>
                  <option value="B" <?php echo $selB;?> >B</option>
                  <option value="C" <?php echo $selC;?> >C</option>
                  <option value="D" <?php echo $selD;?> >D</option>
                  </select>
      </td>
      <td id="error"><?php echo $mCategoria; ?></td>
</tr>
```

Sólo fines educativos - FreeLibros

De la misma forma, para la categoría se imprimirá el mensaje de error en la tercera columna, en caso no seleccione una categoría.

```
<tr>
        <td>Ingrese promedio</td>
        <td><input type="text" name="txtPromedio"
                        value="<?php echo $promedio; ?>" /></td>
        <td id="error"><?php echo $mPromedio; ?></td>
</tr>
```

Y, para finalizar con los mensajes de error, se imprimirá en la tercera columna el mensaje de error del promedio. Un error podría ser ocasionado cuando el usuario deja vacía la caja de ingreso o el valor ingresado no está permitido, como por ejemplo el registro de valores negativos o superiores a veinte.

```
if($categoria=='A')
        $monto=850;
        elseif ($categoria=='B')
        $monto=750;
        elseif ($categoria=='C')
        $monto=650;
        elseif ($categoria=='D')
        $monto=500;
        else
        $monto=0;
```

Usamos la condicional doblemente enlazada para asignar un monto mensual, dependiendo de la categoría seleccionada en la aplicación. Hay que tener en cuenta que el valor por defecto asigna al monto un valor cero, en caso no se haga referencia a ninguna de las cuatro categorías estipuladas en el caso.

```
if ($promedio<=12)
        $descuento=0;
        elseif ($promedio<=15)
        $descuento=10.0/100.0 * $monto;
        elseif ($promedio<=17)
        $descuento=15.0/100.0 * $monto;
        elseif ($promedio<=19)
        $descuento=25.0/100.0 * $monto;
        elseif ($promedio==20)
        $descuento=50.0/100.0 * $monto;

$montoCancelar=$monto-$descuento;
```

Para asignar un monto de descuento también usamos la estructura condicional doblemente enlazada, ello permitirá evaluar los valores obtenidos en el promedio y asignar el descuento según la tabla establecida en el caso. Tenga en cuenta que la condicional doblemente enlazada evalúa y, si encuentra un valor verdadero, sale de la condicional. Tomemos la siguiente fracción de código:

Si el promedio ingresado es 10:

```
if ($promedio<=12)
        $descuento=0;
        elseif ($promedio<=15)
        $descuento=10.0/100.0 * $monto;
        ….
```

Sólo fines educativos - FreeLibros

La condición encuentra verdadero en la primera evaluación y aplica un descuento de 0 %, y ya no sigue evaluando las demás opciones a pesar de que 10 también es menor a 15, menor a 17 y menor a 19.

```
<tr>
    <td>Monto mensualidad</td>
    <td><?php echo '$ '.number_format($monto,2,'.',''); ?></td>
</tr>
<tr>
    <td>Monto descuento</td>
    <td><?php echo '$ '.number_format($descuento,2,'.','');?></td>
</tr>
<tr>
    <td>Monto a cancelar</td>
    <td><?php echo '$ '.number_format($montoCancelar,2,'.',''); ?></td>
</tr>
```

Finalmente, imprimiremos los montos obtenidos en las celdas correspondientes usando la función **number_format** para mostrar en un formato.

Antes de ejecutar la aplicación, debemos agregar un estilo para el error a la lista de estilos iniciales, tal como sigue:

```
#error{
    color: red;
    font-size: 12px;
}
```

- **Caso desarrollado 5: Venta de entradas usando condicional múltiple con switch**

Un teatro desea tener el control de las ventas de sus boletos, según la siguiente tabla:

Días de función	Precio adulto	Precio niño
Lunes	$10.00	$5.00
Martes	$8.00	$4.00
Miércoles a viernes	$16.00	$8.00
Sábado y domingos	$50.00	$45.00

Se necesita implementar una aplicación web con PHP que permita determinar el monto total por las entradas tipo adultos y niños; además, mostrar el nombre del día según la fecha actual; esto se da para la asignación de los precios de las entradas y, finalmente, el monto total a pagar por la compra de ciertas entradas de un determinado comprador.

Finalmente debe considerar:

- La aplicación no deberá mostrar mensajes de error.
- Al iniciar la aplicación web deberá mostrarse la fecha actual en un control de texto, además deberá protegerse de cambios; use el atributo **Readonly=true** del control.
- Deberá usar switch obligatoriamente para la asignación de precios a las diferentes entradas.
- Inicialmente la ventana no debe mostrar los resultados, debe ser como sigue:

Sólo fines educativos - FreeLibros

- Una vez ingresados los valores solicitados, como el nombre del comprador, número de entradas de adultos y niños, se deben mostrar los resultados presionando el botón **Adquirir**, tal como se muestra en la siguiente imagen:

Archivo: **teatro.php**

```
<!DOCTYPE html>
<html>
    <head>
        <meta charset="UTF-8">
        <link href="estilo.css" rel="stylesheet">
        <title>VENTA DE ENTRADAS - TEATRO</title>
    </head>
    <body>
        <header>
```

Sólo fines educativos - FreeLibros

```
        <h2 id="centrado">VENTA DE ENTRADAS (TEATRO)</h2>
        <img src="teatro.jpg" width="600" height="150" />
    </header>
    <section>
    <?php
        error_reporting(0);

        $comprador=$_POST['txtComprador'];
        $fecha=$_POST['txtFecha'];
        $nAdultos=$_POST['txtAdultos'];
        $nNiños=$_POST['txtNiños'];

        $hoy=getdate(time());
        $nDia=$hoy['weekday'];

        switch($nDia){
            case 'Monday': $cAdultos=10; $cNiños=5;break;
            case 'Tuesday': $cAdultos=8; $cNiños=4;break;
            case 'Wednesday':
            case 'Thursday':
            case 'Friday': $cAdultos=16; $cNiños=8;break;
            case 'Saturday':
            case 'Sunday': $cAdultos=50; $cNiños=45;break;
            default : $cAdultos=0; $cNiños=0;
        }

        $adultos=$cAdultos*$nAdultos;
        $niños=$cNiños*$nNiños;
    ?>
    <form name="frmTeatro" method="POST">
        <table border="0" width="600" cellspacing="0"
               cellpadding="0" align="center">
        <tr>
            <td>Comprador</td>
            <td><input type="text" name="txtComprador" size="60"/></td>
        </tr>
        <tr>
            <td>Fecha actual</td>
            <td><input type="text" name="txtFecha" readonly="true"
                       value="<?php echo date('d/m/Y'); ?>" />
            </td>
        </tr>
        <tr>
            <td>Nº entradas adultos</td>
            <td><input type="text" name="txtAdultos" />
            </td>
        </tr>
        <tr>
            <td>Nº entradas niños</td>
            <td><input type="text" name="txtNiños" />
            </td>
        </tr>
        <tr>
            <td></td>
            <td><input type="submit" value="Adquirir"
                       name="btnAdquirir"/></td>
        </tr>
        </table>
        <?php if (isset($_POST['txtComprador'])){ ?>
        <table width="600" border="1" cellspacing="0" cellpadding="0">
        <tr><td>
        <table border="0" width="600" cellspacing="0"
               cellpadding="0" align="center">
```

Sólo fines educativos - FreeLibros

```
            <tr>
                <td width="150">Comprador</td>
                <td width="350"><?php echo $comprador; ?></td>
            </tr>
            <tr>
                <td>Costo por adultos</td>
                <td>
                  <?php echo '$ '.number_format($adultos,2,'.','');?>
                </td>
            </tr>
            <tr>
                <td>Costo por niños</td>
                <td><?php echo '$ '.number_format($niños,2,'.',''); ?></td>
            </tr>
            <tr>
                <td>Dia de promoción</td>
                <td><?php echo $nDia; ?></td>
            </tr>
            <tr>
                <td>Monto total a pagar</td>
                <td>
                  <?php echo '$'.number_format($adultos+$niños,2,'.','');?>
                </td>
            </tr>
            </table>
            </td></tr>
            </table>
            <?php } ?>
        </form>
        </section>
        <footer>
            <h6 id="centrado">Todos los derechos reservados –
                Lic. Manuel Torres</h6>
        </footer>
    </body>
</html>
```

Comentarios:

```
error_reporting(0);

$comprador=$_POST['txtComprador'];
$fecha=$_POST['txtFecha'];
$nAdultos=$_POST['txtAdultos'];
$nNiños=$_POST['txtNiños'];
```

Omitimos los errores de advertencia con la sentencia **error_reporting(0)**, luego obtenemos los valores obtenidos en los controles del formulario. Hay que tener en cuenta que la fecha es tratada como una variable de cadena.

```
$hoy=getdate(time());
$nDia=$hoy['weekday'];

switch($nDia){
        case 'Monday': $cAdultos=10; $cNiños=5;break;
        case 'Tuesday': $cAdultos=8; $cNiños=4;break;
        case 'Wednesday':
        case 'Thursday':
        case 'Friday': $cAdultos=16; $cNiños=8;break;
```

Sólo fines educativos - FreeLibros

```
        case 'Saturday':
        case 'Sunday': $cAdultos=50; $cNiños=45;break;
        default : $cAdultos=0; $cNiños=0;
}

$adultos=$cAdultos*$nAdultos;
$niños=$cNiños*$nNiños;
```

Para determinar los costos, primero debemos capturar la fecha actual, para lo cual usamos la función **getdate(time())**; desde aquí se desprende el nombre del día actual en inglés, con la sentencia **$hoy['weekday']**, la variable **$hoy** almacena un arreglo con información de la fecha actual; desde aquí se desprende el nombre del mes con el parámetro **weekday**.

Una vez obtenido el nombre del mes, se evalúa con la estructura switch; en el caso de lunes (*monday*), se le asigna el costo de adultos ($cAdultos) y costo de niños ($cNiños), tenga en cuenta que cada opción debe contar con la instrucción **break** para culminar, solo se obviará la instrucción **break** en la última opción o la opción por defecto.

Finalmente, se calcula el costo a pagar por el número de entradas de adulto almacenado en la variable $adultos; de la misma forma para el costo a pagar por las entradas de niños en la variable $niños.

```
<tr>
   <td>Fecha actual</td>
   <td><input type="text" name="txtFecha" readonly="true"
           value="<?php echo date('d/m/Y'); ?>" />
   </td>
</tr>
```

Al iniciar la aplicación se debe mostrar la fecha actual, la cual es enviada a la caja de texto **txtFecha**, para lo cual aplicamos el atributo **readonly="true"**, para que el usuario no modifique la fecha mostrada; luego, en la propiedad **value**, imprimimos la fecha actual con la función echo del PHP.

```
<?php if (isset($_POST['txtComprador'])){ ?>
     <table width="600" border="1" cellspacing="0" cellpadding="0">
```

Una de las condiciones del caso especifica que la información resultante solo debe mostrarse si se ingresó el nombre del comprador; caso contrario, no mostrar nada. Para esta condición usamos la función **isset**, que permite evaluar si la variable tiene contenido o no. Observe que se abre la llave antes de iniciar la implementación de la tabla <table>, es así que la llave se cerrará una línea después de la etiqueta de cierre de la tabla </table>.

```
<tr>
        <td>Costo por adultos</td>
        <td><?php echo '$ '.number_format($adultos,2,'.','');?></td>
</tr>
```

El valor obtenido se imprimirá en las celdas correspondientes de la segunda tabla implementada, recuerde que esta información solo se mostrará al ingresar un valor en la caja de texto que almacena el nombre del comprador.

Sólo fines educativos - FreeLibros

CAP.

5

Estructuras repetitivas

Sólo fines educativos - FreeLibros

Sólo fines educativos - FreeLibros

5.1 OPERADORES DE CONTEOS Y ACUMULACIONES

Los operadores de incremento y decremento cumplen diferentes funciones dentro de un código de programación, pues en ocasiones pueden realizar conteos o acumulaciones, también se usa en las estructuras repetitivas como parte de su implementación, es decir, como parte de su formato.

5.1.1 Operadores de incremento y decremento

Permiten aumentar o disminuir en un valor a una determinada variable, normalmente se usa para conteos.

OPERADOR	DESCRIPCIÓN	EJEMPLO
++$variable	También llamada preIncremento, que aumenta el valor de la variable en uno y luego devuelve el valor ya aumentado.	++$n;
$variable++	También llamado postIncremento, en la cual primero devuelve el valor de la variable y luego aumenta en uno.	$n++;
--$variable	También llamado preDecremento, el cual disminuye en uno el valor de la variable y luego devuelve el valor ya disminuido.	--$n;
$variable--	También llamado postDecremento, el cual devuelve el valor de la variable y luego decrementa en uno su valor.	$n--;

5.1.2 Operadores complejos

Permite acumular valores que provienen de sumas, restas, multiplicaciones o divisiones sucesivas, su uso es similar a los contadores y acumuladores.

OPERADOR	DESCRIPCIÓN	EJEMPLO	REFERENCIA
+=	Operador complejo de aumento y acumulador de valores.	$n+=1; (Conteo) $a+=$n; (Acumulador)	$n = $n+1; $a = $a+$n;
-=	Operador complejo de decremento y acumulador de dichos valores.	$n-=1; $a-=$n;	$n = $n-1; $a = $a-$n;
*=	Operador complejo de acumulador de valores en producto.	$a*=$n;	$a = $a*$n;
/=	Operador complejo de acumulador de valores en división.	$a/=$n;	$a = $a/$n;

Sólo fines educativos - FreeLibros

5.2 Contadores

Un contador es una variable normalmente entera en PHP, que permite tener el control de las actividades un número determinado de veces.

Características:

- Casi siempre el contador aumenta o disminuye en uno su valor.
- Las estructuras repetitivas incluyen un contador como parte importante de su implementación.

Formato: $contador = $contador + 1

Donde **contador+1** es la expresión que asigna un valor resultante a la misma variable contador, actualizando así el valor que antes contenía dicha variable.

Veamos el siguiente caso para comprender cómo trabajan los contadores, si tenemos los siguientes billetes de $50, la pregunta sería ¿Cuántos billetes de 50 dólares hay?:

Contador	Valor actualizado	Representación del total del dinero
$cBillete = $cBillete + 1 $cBillete+=1 $cBillete++;	$cBillete = 0+1 1	
$cBillete = $cBillete + 1 $cBillete+=1 $cBillete++;	$cBillete = 1+1 2	
$cBillete = $cBillete + 1 $cBillete+=1 $cBillete++;	$cBillete = 2+1 3	
$cBillete = $cBillete + 1 $cBillete+=1 $cBillete++;	$cBillete = 3+1 4	

Normalmente cuando se realizan conteos se debe inicializar dicha variable con el valor cero; entonces, para este caso asumiremos que el programador lo realizará. Finalmente, la variable **cBillete** tendrá el valor cuatro, por el valor de conteo realizado en los billetes anteriores.

Sólo fines educativos - FreeLibros

5.3 ACUMULADORES

Un acumulador tiene las mismas características del contador con la diferencia de que no cuenta sino acumula los valores.

Características:

- El acumulador aumenta o disminuye su valor a partir de otro valor que no necesariamente es constante.
- Para poder ejecutar de manera correcta un acumulador, siempre debe estar incluido en una estructura repetitiva.

Formato: $acumulador = $acumulador + $valor

Donde $acumulador+$valor es la expresión que asigna un valor resultante a la misma variable acumulador, actualizando así el valor que antes contenía dicha variable.

Veamos el siguiente caso para comprender cómo trabajan los acumuladores, si tenemos los siguientes billetes de $50, la pregunta sería ¿Cuánto dinero tenemos en total?:

ACUMULADOR	VALOR ACTUALIZADO	REPRESENTACIÓN DEL TOTAL DEL DINERO
$aBillete = $aBillete + 50 $aBillete +=50	$aBillete = 0+50 50	
$aBillete = $aBillete + 50 $aBillete+=50	$aBillete = 50+50 100	
$aBillete = $aBillete + 50 $aBillete+=50	$aBillete = 100+50 150	
$aBillete = $aBillete + 50 $aBillete+=50	$aBillete = 150+50 200	

Cuando se realiza una acumulación, la variable debe inicializar con el valor cero; entonces, para este caso asumiremos que el programador lo realizará. Finalmente, la variable **aBillete** tendrá el valor doscientos, por la acumulación de los billetes de $50.00.

Sólo fines educativos - FreeLibros

5.4 Estructura while

Ejecuta repeticiones de una sentencia o un grupo de sentencias un número determinado de veces, según la condición lógica implementada; es decir, esta condición controlará la secuencia de repeticiones antes de que se ejecute.

Formato:

```
while (condición){
    sentencias_repetidas;
}
```

Formato gráfico:

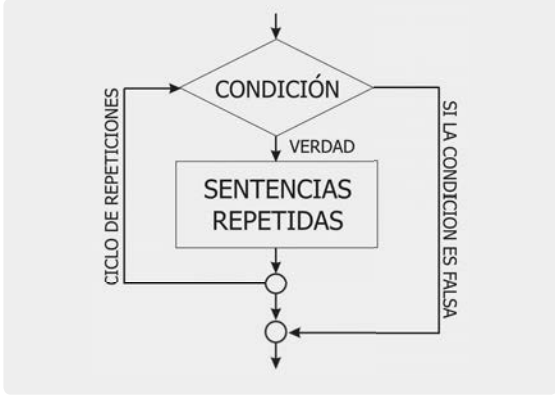


Fig. 5.1 Estructura repetitiva while

Observaciones:

- Considere que la condición lógica usa los operadores lógicos y relacionales, tal como lo usa la condición if.
- La condición es evaluada, y si el resultado es **true** se ejecutarán las sentencias pertenecientes al ciclo de repeticiones, única y exclusivamente si la condición sigue resultando **true**; algunos suelen llamar a esta acción **bucle**.
- Si la condición evaluada resulta **false**, se ejecutarán las sentencias encontradas fuera de la implementación de la estructura **while**.

Veamos algunos ejemplos:

- Imprimir los 10 primeros números en forma ascendente:

```
<?php
    $i=1;
    while($i<=10){
        echo $i.'<br>';
        $i++;
    }
?>
```

Sólo fines educativos - FreeLibros

- Imprimir los 10 primeros números en forma descendente:

```
<?php
    $i=10;
    while($i>=1){
        echo $i.'<br>';
        $i--;
    }
?>
```

- Listar los N primeros pares:

```
<?php
    //Capturamos en valor n
    $n=$_POST['txtN'];

    //Ciclo de repeticiones hasta N
    $i=1;
    while($i <= $n){
        //Buscando los pares
        if($i % 2 == 0)
            echo $i.'<br>';
        $i++;
    }
?>
```

- Listar los N primeros elementos de la siguiente serie: 1/2, 3/4, 5/6, 7/8...N

```
<?php
    //Capturamos en valor N
    $n=$_POST['txtN'];

    //Ciclo de repeticiones hasta N
    $i=1;
    $num=1;
    $den=2;
    while($i <= $n){
        echo $num.'/'.$den.'<br>';
        $num+=2;
        $den+=2;
        $i++;
    }
?>
```

- Listar los N primeros elementos de la siguiente serie: 1/5, -2/10, 3/15, -4/20...N

```
<?php
    //Capturamos en valor N
    $n=$_POST['txtN'];

    //Ciclo de repeticiones hasta N
    $i=1;
    $num=1;
    $den=5;
    while($i <= $n){
        if ($i % 2 != 0)
            echo $num.'/'.$den.'<br>';
            else
```

Sólo fines educativos - FreeLibros

```
                echo '-'.$num.'/'.$den.'<br>';
            $num+=1;
            $den+=5;
            $i++;
        }
    ?>
```

- Imprimir la tabla de multiplicar de un número N:

```
<?php
    //Capturamos en valor N
    $n=$_POST['txtN'];

        //Ciclo de repeticiones
    $i=1;
    while($i <= 12){
        echo $n.'x'.$i.'='.$n*$i.'<br>';
        $i++;
    }
?>
```

5.4.1 Ciclo de repeticiones while con cero iteración

Se le llama así cuando la condición especificada en la estructura **while** resulta **false** en la primera evaluación, como por ejemplo:

```
$i=10;
while($i <= 1){
    echo $i;
    $i--;
}
```

Si la variable i tiene como valor inicial el número 10 cuando esta entra en la evaluación condicional $i<=1; es decir, 10<=1 resulta **false** y, finalmente, se termina el ciclo de repeticiones con cero iteración.

La solución a la cero iteración es usar de la mejor manera posible los operadores de comparación, ya que dependerá de ellos el resultado **true** o **false** de la condición. Entonces la forma adecuada sería:

```
$i=10;
while($i >= 1){
    echo $i;
$i--;
}
```

5.4.2 Ciclo de repeticiones while infinito

Se le llama así cuando el resultado de evaluar la condición resulta todo el tiempo **true**, esto genera un ciclo infinito de repeticiones, como por ejemplo:

```
$i=1;
while($i <= 10){
    echo $i;
}
```

Sólo fines educativos - FreeLibros

Si la variable **i** tiene como inicial el número 1, cuando esta es evaluada por primera vez el resultado es **true** e imprime dicho valor; luego se vuelve a evaluar la condición, resultando nuevamente **true** ya que el valor 1 nunca cambió; imaginemos cuántas vueltas de ciclo se ejecutará, ya que 1 siempre será menor a 10. A esto se le llama while infinito, la forma adecuada sería:

```
$i=1;
while($i <= 10){
     echo $i;
     $i++;
}
```

5.4.3 Uso de la instrucción break en la estructura while

La instrucción **break** permite terminar el ciclo de repeticiones implementada en una estructura while. La idea principal es asignar un **break** dentro de una condición de la estructura selectiva if, veamos un ejemplo básico:

```
<?php
    $n=1000;
    $c=0;
    $i=1;

    while($i<=$n){
        if ($i % 7 == 0){
            echo $i.'<br>';
            $c++;
        }
        if ($c==7) break; else $i++;
    }
?>
```

El código anterior muestra los 7 primeros números múltiplos de 7. Podemos asumir que entre 1 y 1000 se encuentran los 7 primeros múltiplos de 7, es por eso que asignamos 1000 a la variable n, luego necesitamos de un contador llamado **c**, que tendrá la misión de contabilizar la cantidad de veces que encuentra un múltiple. Si este resulta 7, entonces se aplicará la instrucción **break**, caso contrario se continuará con el ciclo de repeticiones.

5.4.4 Uso de la instrucción continue en la estructura while

La instrucción **continue** hace un salto dentro del ciclo de repeticiones, y realiza una nueva evaluación de la condición, veamos un ejemplo básico:

```
<?php
    $n=1000;
    $i=0;
    while($i<=$n){
        $i++;
        if ($i % 7==0)
            continue;
        echo $i.'<br>';
    }
?>
```

Sólo fines educativos - FreeLibros

El código anterior muestra los primeros 1000 números enteros, a excepción de los múltiplos de 7. Para lo cual inicializamos la variable **n** con el valor 1000, la estructura while ejecutará alrededor de 1000 ciclos de repeticiones, de las cuales omitiremos la impresión de los números múltiplos de 7 con la condición **if** ($i%7==0); si el resultado es **true**, entonces aplicamos la instrucción **continue** para que evalúe al siguiente número y omita la impresión de los múltiplos de 7.

5.4.5 Anidamiento de ciclos while

Se le llama así cuando dentro de la estructura while se implementa otro while, de tal manera que trabajan en comunión para la muestra de sus resultados, veamos un ejemplo básico:

```
<?php
    $n=10;
    $i=1;
    while($i <= $n){
        $j=1;
        while($j <= $i){
            echo $i.'  ';
            $j++;
        }
        echo '<br>';
        $i++;
    }
?>
```

En el código anterior se muestra la pirámide de un número entero; es decir, de acuerdo al número impreso se mostrará la cantidad de veces que representa. La salida debe ser como sigue:

```
1
2 2
3 3 3
4 4 4 4
5 5 5 5 5
6 6 6 6 6 6
7 7 7 7 7 7 7
8 8 8 8 8 8 8 8
9 9 9 9 9 9 9 9 9
10 10 10 10 10 10 10 10 10 10
```

5.5 Estructura for

Ejecuta repeticiones de una sentencia o un grupo de sentencias un número fijo de veces.

Formato:

```
for (inicialización; condicion_logica; incremento){
    sentencias_repetidas;
}
```

Sólo fines educativos - FreeLibros

Formato gráfico:

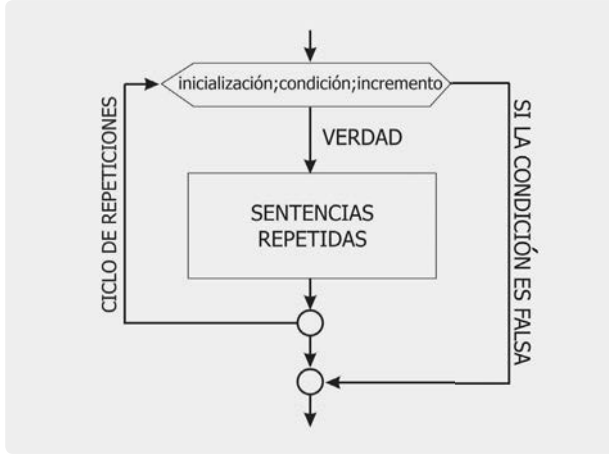


Fig. 5.2 Estructura repetitiva for

Observaciones:

- El valor de inicialización representa el punto de partido del ciclo de repeticiones.
- Considere que la condición lógica usa los operadores lógicos y relacionales, tal como lo usa la condición **if**.
- La condición es evaluada y, si el resultado es **true**, se ejecutarán las sentencias pertenecientes al ciclo de repeticiones únicas, y exclusivamente si la condición sigue resultando **true**; algunos suelen llamar a esta acción **bucle**.
- Si la condición evaluada resulta **false**, se considera como el fin del ciclo de repeticiones; luego se ejecutarán las sentencias encontradas fuera de la implementación de la estructura **for**.

Veamos algunos ejemplos:

- Imprimir los 10 primeros números en forma ascendente:

```
<?php
    for($i=1;$i<=10;$i++)
        echo $i.'<br>';
?>
```

- Imprimir los 10 primeros números en forma descendente:

```
<?php
    for($i=10;$i>=1;$i--)
        echo $i.'<br>';
?>
```

Sólo fines educativos - FreeLibros

- Listar los N primeros pares:

```
<?php
    //Capturamos en valor n
    $n=$_POST['txtN'];

    //Ciclo de repeticiones hasta N
    for($i=1;$i<=$n;$i++){
        //Buscar los pares
        if($i % 2 ==0)
            echo $i. '<br>';
    }
?>
```

- Listar los N primeros elementos de la siguiente serie: 2/1, 4/2, 6/3, 8/4...N y determinar la suma de sus elementos.

```
<?php
    //Capturamos en valor N
    $n=$_POST['txtN'];

    $num=2;
    $den=1;
    $suma=0;
    for($i=1;$i<=$n;$i++){
       echo $num.'/'.$den.'<br>';
       $suma+=$num/$den;
       $num+=2;
       $den+=1;
    }
    echo 'La suma es: '.$suma;
?>
```

- Listar los divisores de un determinado número entero:

```
<?php
    //Capturamos en valor N
    $n=$_POST['txtN'];

    echo 'Los divisores del numero '.$n.' es:<br>';
    for($i=1;$i<=$n;$i++){
       if ($n % $i ==0)
           echo $i.'<br>';
    }
?>
```

- Imprimir la tabla de multiplicar de un número N:

```
<?php
    //Capturamos en valor N
    $n=$_POST['txtN'];

        for($i=1;$i<=12;$i++)
                    echo $n .'x'. $i .'='. $n*$i .'<br>';
?>
```

Sólo fines educativos - FreeLibros

5.5.1 Analogías entre while y for

Cuando se implementan una estructura while o for, el resultado siempre será el mismo, el uso de uno u otro dependerá propiamente del desarrollador; por eso haremos una analogía entre códigos usando while y for en una determinada situación:

- Listar los 10 primeros números en forma ascendente

```
$i=1;
while ($i<=10){
 echo $i;
 $i++;
}
```

```
for ($i=1;$i<=10;$i++){
 echo $i;
}
```

- Listar los 10 primeros números en forma descendente

```
$i=10;
while ($i>=1){
 echo $i;
 $i--;
}
```

```
for ($i=10;$i>=1;$i--){
 echo $i;
}
```

5.5.2 Uso de la instrucción break en la estructura for

La instrucción **break** permite finalizar el ciclo de repeticiones, hay que tener en cuenta que dicho corte debe implementarse dentro de una estructura condicional **if**, por ejemplo:

```
<?php
    $n=1000;
    $c=0;

    for($i=1;$i<=$n;$i++){
        if ($i % 5 == 0){
            echo $i.'<br>';
            $c++;
        }
        if ($c==10) break;
    }
?>
```

El código anterior permite imprimir los 10 primeros números múltiplos de 5, aquí la estructura **for** solo se encarga de recorrer hasta el número 1000; internamente se condiciona a que dicha variable sea divisible por 5. Si resulta **true**, entonces se imprimirá y se contará cuantas veces se encontró dicho valor; fuera de la condición se preguntará si la variable **c** ya tiene valor 10, para lo cual finalizará el ciclo de repeticiones con la instrucción **break**, ya que habremos encontrado el décimo número múltiplo de 5.

Sólo fines educativos - FreeLibros

5.5.3 Uso de la instrucción continue en la estructura for

La instrucción **continue** hace un salto dentro del ciclo de repeticiones y realiza una nueva evaluación de la condición, veamos un ejemplo básico:

```
<?php
    $n=1000;
    for($i=1;$i<=$n;$i++){
        if ($i % 5==0)
            continue;
        echo $i.'<br>';
    }
?>
```

El código anterior muestra los primeros 1000 números enteros a excepción de los múltiplos de 5. Para lo cual inicializamos la variable **n** con el valor 1000, la estructura **for** ejecutará alrededor de 1000 ciclos de repeticiones de las cuales omitiremos la impresión de los números múltiplos de 5 con la condición **if** ($i%5==0); si el resultado es **true**, entonces aplicamos la instrucción **continue** para que evalúe al siguiente número y omita la impresión de los múltiplos de 5.

5.5.4 Anidamiento de ciclos for

Se le llama así cuando dentro de la estructura **for** se implementa otro **for**, de tal manera que trabajan en comunión para la muestra de sus resultados, veamos un ejemplo básico:

```
<?php
    $n=1000;
    $t=0;
    echo 'Los 10 primeros primos son: <br>';
    for($i=1;$i<=$n;$i++){
        $c=0;
        for($j=1;$j<=$i;$j++){
            if ($i % $j == 0){
                $c++;
            }
        }
        if ($c==2){
            echo $i.'<br>';
            $t++;
        }
        if($t==10) break;
    }
?>
```

En el código anterior se muestra los 10 primeros números primos, para lo cual inicializamos el valor 1000 a una variable **n**; asumiendo que entre 1 y 1000 se encuentran los 10 primeros primos, luego se inicializa la variable **t** con el valor cero, este determinará el número de veces que encontramos a un número primo.

Se inicia el ciclo de repeticiones para la variable **i**, el cual recorre desde 1 hasta 1000, y dentro de este se inicia el otro ciclo de repeticiones; pero esta vez para la variable **j** que será la encargada de recorrer desde el valor 1 hasta el valor que por ciclo de repeticiones obtenga **i**, veamos un cuadro para explicar hasta este punto:

Sólo fines educativos - FreeLibros

i	1	2	3	4	5	6	7	8	9	10	11	12	13	J
	1	1	1	1	1	1	1	1	1	1	1	1	1	
		2	3	2	5	2	7	2	3	2	11	2	13	
				4		3		4	9	5		3		
						6		8		10		4		
												6		
												12		

Si observamos en el cuadro, la variable **i** es la responsable de recorrer los números en forma consecutiva; es decir, su límite será el valor 1000, mientras que la variable **j** iniciará el ciclo de repeticiones en el valor 1 y terminará en el valor contenido en **i**; es decir, si **i** tiene el valor 8 la variable **j** tendrá su valor de inicio en 1 y terminará en 8.

Si **j** es divisible por **i**, contaremos cuántas veces sucede lo mismo, ya que según la tabla todo número que tiene dos divisores es catalogado como primo. Hay que tener en cuenta que usamos un contador para dicho conteo y debe inicializarse por cada ciclo de repeticiones de **i**. Si dicho contador es igual a 2, se imprimirá la variable **i**, y a la vez aumentamos en uno a la variable **t** para conseguir el límite de primos a mostrar.

Finalmente, si la variable **t** es igual a 10; es decir, ya mostro los 10 primeros primos, entonces se corta el ciclo de repeticiones con la instrucción **break**.

5.6 ESTRUCTURA DO... WHILE

La forma de trabajo es similar a la estructura **while** y **for** con la diferencia de que por lo menos se ejecuta las instrucciones al menos una vez.

Formato:

```
do{
        sentencias_repetidas;
} while(condición_logica);
```

Se dice que se ejecuta por lo menos una vez las sentencias repetitivas, ya que la condición se encuentra al final de la instrucción. Hay que considerar que no se modifica la forma de trabajo de la condición; es decir, para continuar en el ciclo de repeticiones, la condición lógica debe emitir siempre **true**; cuando es **false** termina el ciclo de repeticiones.

Formato gráfico:

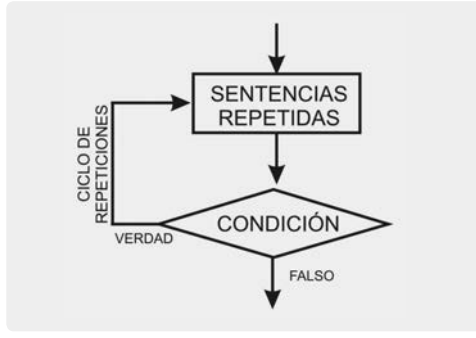


Fig. 5.3 Estructura repetitiva do...while

Sólo fines educativos - FreeLibros

5.6.1 Analogías entre while, for y do...while

Ahora dependerá del desarrollador el uso de las estructuras repetitivas, pues cada una de ellas presenta formato distinto, pero siempre generan ciclos de repeticiones:

- Listar los 10 primeros números en forma ascendente

```
$i=1;
while ($i<=10){
 echo $i;
 $i++;
}
```

```
for ($i=1;$i<=10;$i++){
 echo $i;
}
```

```
$i=1;
do{
  echo $i.'<br>';
  $i++;
}while($i<=10);
```

- Listar los 10 primeros números en forma descendente

```
$i=10;
while ($i>=1){
 echo $i;
 $i--;
}
```

```
for ($i=10;$i>=1;$i--){
 echo $i;
}
```

```
$i=10;
do{
  echo $i.'<br>';
  $i--;
}while($i>=1);
```

5.7 Casos desarrollados

Para la implementación de los casos desarrollados usaremos el siguiente estilo:

Archivo: **estilo.css**

```
body{
    font-family: tahoma;
    font-size: 14px;
}
#centrado{
    text-align: center;
}
table {
    margin: auto;
}
img{
    margin: auto;
    display: block;
}
#fila{
    font-size: 13px;
    background: #b9c9fe;
    color: #039;
}
#error{
    color: red;
    font-size: 12px;
}
```

Sólo fines educativos - FreeLibros

Caso desarrollado 1: Venta de productos usando while

Una tienda comercial necesita implementar una aplicación web que permita vender un determinado producto; esta venta se realizará al crédito, por lo cual el cliente podrá seleccionar el número de cuotas. Y en base al monto subtotal de la venta, se obtendrá una lista de letras con los montos a pagar.

La interfaz gráfica inicial es la siguiente:

Tener en cuenta:

- La aplicación no deberá mostrar mensajes de error.
- Los productos ya deberán estar precargados en el control SELECT.
- Al seleccionar un producto, deberá mostrar el precio en el control cuadro de texto.
- Deberá ingresar la cantidad del producto a comprar, y para imprimir el subtotal debe seleccionar el botón **Calcular**.
- Al seleccionar un determinado número de cuotas, deberá calcular los montos según la cuota seleccionada.
- Los productos y sus precios se presentan en la siguiente tabla:

PRODUCTO	PRECIO
Lavadora	$ 1500.00
Refrigeradora	$ 3500.00
Radiograbadora	$ 500.00
Tostadora	$ 150.00

Sólo fines educativos - FreeLibros

- Los valores resultantes deben mostrarse como la siguiente imagen:

Archivo: **venta.php**

```
<!DOCTYPE html>
<html>
    <head>
        <meta charset="UTF-8">
        <title>Venta de Productos</title>
        <link href="estilo.css" rel="stylesheet">
    </head>
    <body>
      <header>
       <img src="bannerElectrodomesticos.jpg" width="780"
            height="169" alt="bannerElectrodomesticos"/>
       <h2 id="centrado">VENTA DE PRODUCTOS</h2>
      </header>
      <?php
        error_reporting(0);

        $producto=$_POST['selProducto'];

        $precio=0;
        switch ($producto){
         case 'Lavadora': $precio=1500; break;
         case 'Refrigeradora': $precio=3500; break;
         case 'Radiograbadora': $precio=500; break;
         case 'Tostadora': $precio=150;
        }

        if ($producto=='Lavadora') $selL='SELECTED'; else $selL="";
        if ($producto=='Refrigeradora') $selRe='SELECTED'; else $selRe="";
        if ($producto=='Radiograbadora') $selRa='SELECTED'; else $selRa="";
        if ($producto=='Tostadora') $selT='SELECTED'; else $selT="";
        ?>

        <form method="POST" name="frmVenta">
          <table border="0" width="500" cellspacing="0" cellpadding="0">
```

Sólo fines educativos - FreeLibros

```
<tr>
  <td>Producto</td>
  <td><select name="selProducto" onchange="this.form.submit()">
          <option value="Lavadora" <?php echo $selL;?>>
               Lavadora</option>
          <option value="Refrigeradora" <?php echo $selRe;?>>
               Refrigeradora</option>
          <option value="Radiograbadora" <?php echo $selRa;?>>
               Radiograbadora</option>
          <option value="Tostadora" <?php echo $selT;?>>
               Tostadora</option>
      </select>
   </td>
</tr>
<tr>
 <td>Precio</td>
 <td>
   <input type="text" name="txtPrecio" readonly="readonly"
     value="<?php
              if ($_POST[selProducto])
                 echo number_format($precio,'2','.','');
            ?>" />
 </td>
</tr>
<?php
    $cantidad=$_POST['txtCantidad'];
    $subtotal=$cantidad*$precio;
?>
<tr>
  <td>Cantidad</td>
  <td>
    <input type="text" name="txtCantidad"
          value="<?php echo $cantidad; ?>" />
  </td>
  <td>
    <input type="submit" value="Calcular" name="btnCalcular" />
  </td>
 </tr>

  <tr>
    <td>Subtotal</td>
    <td><input type="text" name="txtSubtotal"
         value="<?php
              echo '$'.number_format($subtotal,'2','.',''); ?>"
        />
    </td>
    </tr>
    <tr>
        <td>Cuotas</td>
        <td>
          <select name="selCuotas"
                  onchange="this.form.submit()">
                <option value="3">3</option>
                <option value="6">6</option>
                <option value="9">9</option>
                <option value="12">12</option>
          </select>
        </td>
    </tr>
    <?php
        if ($_POST['selCuotas']) {
```

Sólo fines educativos - FreeLibros

```
                ?>
                <tr id="fila">
                    <td>Nº Letras</td>
                    <td>Monto</td>
                </tr>
                <?php
                    $cuotas=$_POST['selCuotas'];
                    $i=1;
                    $montoMensual = $subtotal/$cuotas;
                    while($i<=$cuotas){
                ?>
                <tr>
                 <td><?php echo $i; ?></td>
                 <td>
                  <?php echo '$'.number_format($montoMensual,'2','.','');?>
                 </td>
                </tr>
                <?php
                    $i++;
                    }
                  }
                ?>
            </table>
        </form>
    </body>
</html>
```

Comentarios:

```
error_reporting(0);

$producto=$_POST['selProducto'];

$precio=0;
switch ($producto){
        case 'Lavadora': $precio=1500; break;
        case 'Refrigeradora': $precio=3500; break;
        case 'Radiograbadora': $precio=500; break;
        case 'Tostadora': $precio=150;
}
```

Omitimos los errores de advertencia, luego capturamos el producto seleccionado el cuadro combinado de los productos.

Asignaremos los precios de los productos usando la estructura switch, pero antes debemos asignar el precio con el valor cero como iniciador para evitar errores de precisión. Luego, según el producto seleccionado por el usuario, se asignará un precio; esto dependerá de la tabla propuesta en el caso, no se olvide que por cada opción debe asignar un **break**.

```
if ($producto=='Lavadora') $selL='SELECTED'; else $selL="";
if ($producto=='Refrigeradora') $selRe='SELECTED'; else $selRe="";
if ($producto=='Radiograbadora') $selRa='SELECTED'; else $selRa="";
if ($producto=='Tostadora') $selT='SELECTED'; else $selT="";
```

Para mantener en el cuadro combinado el producto seleccionado debemos asignar la opción SELECTED a la opción seleccionada por el usuario, luego se asignarán estas variables a cada opción del menú.

Sólo fines educativos - FreeLibros

```
<td><select name="selProducto" onchange="this.form.submit()">
                        <option value="Lavadora" <?php echo $selL;?>>
                             Lavadora</option>
                        <option value="Refrigeradora" <?php echo $selRe;?>>
                             Refrigeradora</option>
                        <option value="Radiograbadora" <?php echo $selRa;?>>
                             Radiograbadora</option>
                        <option value="Tostadora" <?php echo $selT;?>>
                             Tostadora</option>
     </select>
</td>
```

Hay que considerar que el precio del producto no será ingresado por el usuario, esto se obtendrá a partir de la selección de un producto desde el cuadro combinado; para esto debemos habilitar la opción SUBMIT del cuadro combinado, para ello debemos habilitar la propiedad onchange="this.form.submit()" en el menú **selProducto**, esto permitirá accionar el cuadro combinado como si fuera un botón tipo **submit**.

No olvide que teníamos variables para cada opción, las cuales permitían establecer la opción por defecto del menú de opción, para que se pueda mantener el producto seleccionado en el mismo control. Eso fue impreso en cada opción con la función **echo** de PHP, tal como se muestra en el siguiente código <option value="Lavadora" <?php echo $selL;?>>.

```
<td>
  <input type="text" name="txtPrecio" readonly="readonly"
         value="<?php
                   if ($_POST[selProducto])
                        echo number_format($precio,'2','.','');
                ?>" />
</td>
```

Según la condición del caso, solo debe mostrarse el precio cuando se ha seleccionado por lo menos un producto de la lista, esto se evalúa con if ($_POST[selProducto]), el cual evalúa dicha capacidad. Seguidamente, si es correcta la selección, se mostrará el precio en un celda de la tabla, cambiando en el momento al seleccionar uno u otro producto.

```
<?php
      $cantidad=$_POST['txtCantidad'];
      $subtotal=$cantidad*$precio;
?>
```

Capturamos la cantidad ingresada por el usuario y calculamos en base a dicho valor el subtotal a pagar; tenga en cuenta que la cancelación es al crédito, por tanto, el valor subtotal deberá distribuirse entre las cuotas seleccionadas por el usuario.

```
<td>
      <input type="text" name="txtCantidad" value="<?php echo $cantidad; ?>" />
</td>
```

En el siguiente código se imprime la variable **cantidad** por la propiedad VALUE del control de texto del formulario, esto mantendrá el valor visible de la cantidad.

Sólo fines educativos - FreeLibros

```
<td><input type="text" name="txtSubtotal"
        value="<?php  echo '$'.number_format($subtotal,'2','.',''); ?>" />
</td>
```

En el siguiente código imprimiremos el valor de la variable **subtotal** en la celda correspondiente.

```
<td>
  <select name="selCuotas" onchange="this.form.submit()">
        <option value="3">3</option>
        <option value="6">6</option>
        <option value="9">9</option>
        <option value="12">12</option>
  </select>
</td>
```

Para mostrar los montos y el número de cuotas, debemos habilitar la opción Submit del control cuadro combinado, con la opción onchange=”this.form.submit()”

```
<?php
       if ($_POST['selCuotas']) {
?>
<tr id="fila">
       <td>Nº Letras</td>
       <td>Monto</td>
</tr>
```

Para mostrar los valores resultantes al seleccionar un producto, se deberá comprobar si hay un valor seleccionado en la variable **selCuotas**. Hay que tener en cuenta que las celdas posteriores de las tablas se mostrarán solo si dicha variable tiene valor; caso contrario, no mostrará ni el valor resultante ni las celdas que lo componen.

```
<?php
       $cuotas=$_POST['selCuotas'];
       $i=1;
       $montoMensual = $subtotal/$cuotas;
       while($i<=$cuotas){
?>
```

Ahora capturamos el número de cuotas, desde el control **selCuotas**, para crear una estructura repetitiva con el número total de cuotas y así poder imprimir los montos. La variable **$i** genera el punto de inicio de las cuotas, ya que cualquiera de las cuotas seleccionadas siempre empezará en la cuota uno. Los montos mensuales son calculados en base al subtotal y el número de cuotas.

```
<td><?php echo $i; ?></td>
<td><?php echo '$'.number_format($montoMensual,'2','.','');?></td>
```

Se imprimirá el valor obtenido por **$i**, el cual tiene el número de cuotas y el monto mensual, el cual repetirá el valor obtenido por $montoMensual = $subtotal/$cuotas, cada uno en las celdas correspondientes.

Sólo fines educativos - FreeLibros

```
<?php
        $i++;
        }
    }
?>
```

La variable **$i** debe aumentar en uno para poder imprimir un nuevo valor por cada ciclo de repeticiones, hasta que el número de cuotas llegue al límite, esto lo realizará la estructura while. La primera llave cierra el ciclo de repeticiones del while, mientras que la segunda cierra la estructura condicional **if** ($_POST['selCuotas']).

- **Caso desarrollado 2: Pago de préstamo usando for**

Una casa de préstamo necesita crear una aplicación web que permita simular los montos de las cuotas que un cliente debe realizar; además de mostrar las fechas posibles de pago a partir del mes siguiente.

La interfaz gráfica inicial es la siguiente:

Tener en cuenta:

- La aplicación no deberá mostrar mensajes de error.
- El número de cuotas cuenta con un criterio especial de impuesto, el cual se muestra en el siguiente cuadro.

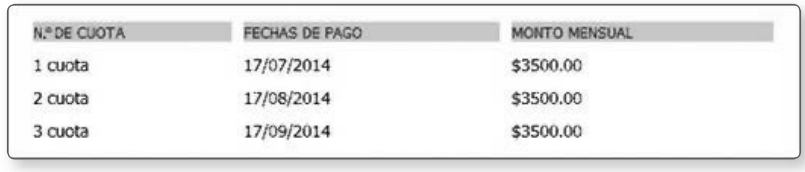
NÚMERO DE CUOTAS	IMPUESTO
3	5 %
6	7 %
9	10 %
12	20 %

Sólo fines educativos - FreeLibros

- Valide el ingreso del nombre del cliente mostrando el mensaje «Debe registrar nombre del cliente», en caso el usuario deje vacía esta sección.
- Valide el ingreso del monto prestado mostrando el mensaje «Debe registrar correctamente el monto de préstamo», solo cuando el usuario deje vacío o ingrese una cadena en el cuadro de texto del monto prestado.
- Use la estructura **for** para la impresión del número de cuotas, fechas de pago y monto mensual, según el número de cuotas seleccionado.
- Inicialmente, la ventana no debe mostrar los resultados hasta que presione el botón **Cotizar**; al mostrar los valores resultantes deberá tener el siguiente aspecto:

- Implemente una tabla de 4 filas por 3 columnas para el siguiente entorno:

- Implemente una segunda tabla de 2 filas por 3 columnas para la impresión de las cuotas, fechas de pago y monto mensual para el siguiente entorno:



Sólo fines educativos - FreeLibros

Archivo: **pago.php**

```
<!DOCTYPE html>
<html>
    <head>
        <meta charset="UTF-8">
        <link href="estilo.css" rel="stylesheet">
        <title>Pago de Prestamo</title>
    </head>
    <body>
     <header>
      <h2 id="centrado">CASA DE PRESTAMO</h2>
      <img src="pago.jpg" width="650" height="180" alt="pago"/>
     </header>
     <section>
      <?php
        error_reporting(0);

        $cliente=$_POST['txtCliente'];
        $monto=$_POST['txtMonto'];
        $cuotas=$_POST['selCuotas'];

        if ($cuotas==3) $sel3='SELECTED'; else $sel3="";
        if ($cuotas==6) $sel6='SELECTED'; else $sel6="";
        if ($cuotas==9) $sel9='SELECTED'; else $sel9="";
        if ($cuotas==12) $sel12='SELECTED'; else $sel12="";

        $mCliente='';
        $mMonto='';
        if (empty($cliente))
          $mCliente='Debe registrar nombre del cliente';
        if(empty($monto) || !is_numeric($monto))
          $mMonto='Debe registrar correctamente el monto de préstamo';
          elseif($monto<=0)
          $mMonto='El monto prestamos no debe ser inferior a 0';
       ?>
       <form method="POST" name="frmPrestamo" action="pago.php">
         <table border="0" width="650" cellspacing="10" cellpadding="0">
           <tr>
             <td width="">Cliente</td>
             <td>
                <input type="text" name="txtCliente" size="70"
                       value="<?php echo $_POST['txtCliente']; ?>" />
             </td>
             <td  width="200" id="error"><?php echo $mCliente; ?> </td>
           </tr>
           <tr>
             <td>Monto Prestado</td>
             <td>
                <input type="text" name="txtMonto"
                       value="<?php echo $_POST['txtMonto']; ?>" />
             </td>
             <td id="error"><?php echo $mMonto; ?></td>
            </tr>
            <tr>
             <td>Nº Cuotas</td>
             <td><select name="selCuotas">
                    <option value="3" <?php echo $sel3;?> >3</option>
                    <option value="6" <?php echo $sel6;?> >6</option>
                    <option value="9" <?php echo $sel9;?> >9</option>
                    <option value="12" <?php echo $sel12;?> >12</option>
```

Sólo fines educativos - FreeLibros

```
                    </select>
                </td>
                <td></td>
              </tr>
              <tr>
                <td></td>
                <td><input type="submit" value="Cotizar" /></td>
              </tr>
            </table>
            <?php if(!empty($cliente) && !empty($monto)) { ?>
            <table border="0" width="650" cellspacing="10" cellpadding="0">
              <tr id="fila">
                <td>Nº DE CUOTA</td>
                <td>FECHAS DE PAGO</td>
                <td>MONTO MENSUAL</td>
              </tr>
              <?php
                switch ($cuotas){
                  case 3: $montoMensual=($monto*1.05)/$cuotas; break;
                  case 6: $montoMensual=($monto*1.07)/$cuotas; break;
                  case 9: $montoMensual=($monto*1.10)/$cuotas; break;
                  case 12: $montoMensual=($monto*1.20)/$cuotas;
                } //Cierre del switch

                $fecha = date('d-m-Y');
                for($i=1;$i<=$cuotas;$i++){
              ?>
              <tr>
                <td><?php echo $i.' cuota';?></td>
                <td>
                  <?php echo date('d/m/Y', strtotime("$fecha +$i month"));?>
                </td>
                <td>
                  <?php echo '$'.number_format($montoMensual,'2','.',''); ?>
                </td>
              </tr>
              <?php } //Cierre del for ?>
            </table>
            <?php } //Cierre del if ?>
              </form>
          </section>
          <footer>
              <h6 id="centrado">Todos los derechos reservados –
                  Lic. Manuel Torres</h6>
          </footer>
      </body>
  </html>
```

Comentarios:

```
error_reporting(0);

$cliente=$_POST['txtCliente'];
$monto=$_POST['txtMonto'];
$cuotas=$_POST['selCuotas'];
```

Omitimos los errores de advertencia, luego asignamos a cada variable el valor ingresado y seleccionado por el usuario, considere $_POST como función de captura de valor.

Sólo fines educativos - FreeLibros

```
if ($cuotas==3) $sel3='SELECTED'; else $sel3='';
if ($cuotas==6) $sel6='SELECTED'; else $sel6='';
if ($cuotas==9) $sel9='SELECTED'; else $sel9='';
if ($cuotas==12) $sel12='SELECTED'; else $sel12='';
```

Asignamos el valor predeterminado en el control SELECT para que al seleccionar una cuota esta se quede impresa en su mismo control. Se ha creado una variable para cada opción, donde, si el número de cuotas es correcta, se le asignará el valor SELECTED, que permite asignar el foco a dicho elemento y, por lo tanto, queda impreso en el control cuadro combinado.

```
$mCliente='';
$mMonto='';

if (empty($cliente))
    $mCliente='Debe registrar nombre del cliente';

if(empty($monto) || !is_numeric($monto))
    $mMonto='Debe registrar correctamente el monto de préstamo';
    elseif($monto<=0)
    $mMonto='El monto prestamos no debe ser inferior a 0';
```

Se declaran **mCliente** y **mMonto** como variables que recibirán el mensaje de error según la naturaleza del problema encontrado; estas deben estar asignadas con un espacio en blanco, es decir, entre comillas simples o dobles. La función **empty** permite determinar si la variable se encuentra o no vacía e **isnumeric** permite determinar si el numero ingresado es numérico o no.

```
<td> <input type="text" name="txtCliente" size="70"
                value="<?php echo $_POST['txtCliente']; ?>" /> </td>

<td> <input type="text" name="txtMonto"
                  value="<?php echo $_POST['txtMonto']; ?>" /> </td>

  <td>
    <select name="selCuotas">
      <option value="3" <?php echo $sel3;?> >3</option>
      <option value="6" <?php echo $sel6;?> >6</option>
      <option value="9" <?php echo $sel9;?> >9</option>
      <option value="12" <?php echo $sel12;?> >12</option>
    </select>
  </td>
```

Para mantener la información ingresada por el usuario en el control **txtCliente**, se le asigna la impresión mediante la función **echo** de PHP al atributo VALUE; de la misma manera se le asignará al control **txtMonto**.

De igual forma se tiene que asignar la impresión mediante **echo** de PHP en el control cuadro combinado, pues existen variables que deben asignadas a cada opción <option value="3" <?php echo $sel3;?> >3</option>.

Sólo fines educativos - FreeLibros

```
<td id="error"> <?php echo $mCliente; ?> </td>
<td id="error"> <?php echo $mMonto; ?> </td>
```

Si consideramos que inicialmente se implementó una tabla de 3 columnas por 4 filas, es en la tercera columna en la que debemos imprimir los mensajes de error ocasionados por algún error de ingreso del usuario, lo que nosotros consideramos como validación. Hay que tener en cuenta que de hallar o no un error, igual se estará imprimiendo un mensaje en la tercera columna.

```
<?php if(!empty($cliente) && !empty($monto)) { ?>
```

Se realiza la validación del nombre de cliente y el monto ingresado, ya que de ellos dependerá la impresión de resultados en la segunda tabla de la aplicación, hay que tener en cuenta que la función **empty** determinar si la variable no tiene contenido, pero al asignarle el símbolo de admiración negamos dicha acción y, por lo tanto, la condición sería «si no está vacío el nombre del cliente ni tampoco el monto» entonces se mostrarán los resultados.

Finalmente imprimiremos los resultados

Nº DE CUOTA	FECHAS DE PAGO	MONTO MENSUAL
1 cuota	17/07/2014	$3500.00
2 cuota	17/08/2014	$3500.00
3 cuota	17/09/2014	$3500.00

```
<?php
for($i=1;$i<=$cuotas;$i++){
?>
        <tr>
        <td><?php echo $i.' cuota';?></td>
        <td>
          <?php echo date('d/m/Y', strtotime("$fecha +$i month"));?>
        </td>
        <td><?php echo '$'.number_format($montoMensual,'2','.',''); ?></td>
        </tr>
<?php } ?>
```

La función **strtotime** permite realizar operaciones sobre una determinada fecha; en el caso de los pagos mensuales se efectuará el aumento en el atributo **month**. No se olvide que debe usar doble comilla para la definición del parámetro “$fecha +$i month”, **$fecha** representa la fecha actual y **$i** representa la cantidad de cuotas; es aquí donde se generan las fechas de pago.

Sólo fines educativos - FreeLibros

CAP.

6

Funciones

Sólo fines educativos - FreeLibros

Sólo fines educativos - FreeLibros

6.1 FUNCIONES PARA VARIABLES

Veamos las principales funciones que tiene PHP para el manejo de las variables; hay que tener en cuenta que las funciones afectan directamente a los valores contenidos en la variable.

6.1.1 Función isset

Devuelve **true** si una variable de cualquier tipo tiene un valor asignado; hay que tener en cuenta que un espacio en blanco es considerado como valor para una variable de tipo cadena. Su formato es:

```
isset($variable)
```

Veamos algunos casos:

- Si necesitamos comprobar si una variable de tipo cadena tiene un valor asignado, podemos usar el siguiente código:

```
<?php
    $titulo='Fundamentos de programación con PHP';
    if (isset($titulo))
        echo 'La variable tiene contenido';
        else
            echo 'La variable no tiene definición';
?>
```

Resultado:

```
La variable tiene contenido
```

Hay que tener en cuenta que si la variable está definida de la siguiente manera **$titulo=''**; también devolverá el mensaje «La variable tiene contenido», pues el espacio también es considerado como un valor.

- Ahora comprobaremos cuál será la salida si evaluamos una variable numérica; debemos tener en cuenta que el resultado nos debe arrojar el tipo de datos y su definición **true** o **false**, según el valor de la variable:

```
<?php
    $resultado = isset($edad);
    echo 'El variable tiene por definición: ';
    echo var_dump($resultado);
?>
```

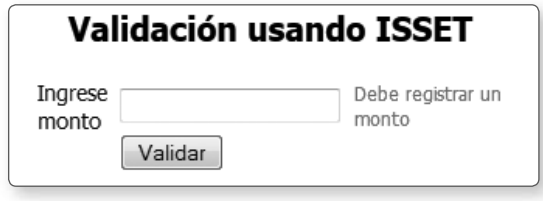
Resultado:

```
El variable tiene por definición:

boolean false
```

Sólo fines educativos - FreeLibros

La función **var_dump** tiene por misión imprimir información acerca de la variable a evaluar; en este caso, evalúa cuál es el resultado del **isset** a la variable edad, que no está definida.

- En el siguiente ejemplo validaremos el ingreso de un valor en un control TEXT, al iniciar la aplicación debe mostrar el mensaje «Debe registrar un monto», tal como se muestra en la siguiente imagen:



Si el usuario ingresa un valor en el control TEXT y presiona el botón **validar**, se mostrará el mensaje «Ingreso correcto», tal como se muestra en la siguiente imagen:

El código PHP para la solución del ejemplo es:

```
<!DOCTYPE html>
<html>
    <head>
        <meta charset="UTF-8">
        <title>Validaciones</title>
        <link href="estilo.css" rel="stylesheet">
    </head>
    <body>
        <header>
            <h2 id="centrado">Validacion usando ISSET</h2>
        </header>
        <section>
        <?php
            error_reporting(0);
            if(isset($_POST['txtMonto']))
                $mensaje='Ingreso correcto';
                else
                    $mensaje='Debe registrar un monto';
        ?>
        <form name="frmValida" method="POST">
            <table border="0" width="300"
                   cellspacing="0" cellpadding="0">
                <tr>
                    <td width="100">Ingrese monto</td>
                    <td width="200">
                      <input type="text" name="txtMonto"
                          value="<?php echo $_POST['txtMonto'];?>"/>
                    </td>
                    <td width="400"><?php echo $mensaje; ?></td>
                </tr>
```

Sólo fines educativos - FreeLibros

```
                    <tr>
                        <td></td>
                        <td><input type="submit" value="Validar" /></td>
                        <td></td>
                    </tr>
                </table>
            </form>
            </section>
        </body>
    </html>
```

Donde **if** (isset($_POST['txtMonto'])) evalúa si se ingresó algún valor en el control TEXT, de acuerdo al que se ingrese se asignará uno u otro comentario a la variable **$mensaje** y esta se imprimirá en la celda contigua al control TEXT del monto.

Hay que considerar que la evaluación realizada solo controla si la variable tiene algún valor, no necesariamente numérico.

Finalmente, el código PHP se apoya del archivo **estilo.css**, que permite presentar de la mejor manera posible al formulario:

```
body{
    font-family: tahoma;
    font-size: 14px;
}

#centrado{
    text-align: center;
}
table {
    margin: auto;
}
#error{
    color: red;
    font-size: 12px;
}
```

6.1.2 Función unset

Permite borrar de la memoria de la computadora una variable que contenía algún valor, dejándola inactiva e irreconocible por la aplicación, su formato es:

```
unset($variable);
unset($variable1, $variable2, $variable3,…);
```

Tener en cuenta:

- La función **unset** destruye a una o más variables.
- No se considera igual que asignar **null** a una variable.
- Cuando se aplica **unset** dentro de una función o método definido por el usuario, destruirá las variables locales de la función; es decir, dentro de la función dicha variable no será utilizable.
- Si se necesita destruir una variable global se debe seguir el siguiente formato:

```
unset($GLOBALS['variable']);
unset($GLOBALS['variable1'], $GLOBALS['variable2'],…);
```

Sólo fines educativos - FreeLibros

6.1.3 Función gettype

Permite obtener el tipo de datos de una determinada variable. Su formato es:

```
$variable=gettype($variable);
```

Los tipos de datos que puede devolver la función **gettype** son:

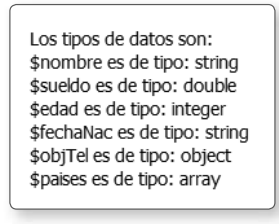
- Si es lógico el valor, entonces devolverá boolean.
- Si es un número entero, devolverá integer.
- Si es un número fraccionario, devolverá double.
- Si es una cadena de caracteres, devolverá string.
- Si es una colección de valores, devolverá array.
- Si es una instancia de una clase, entonces devolverá object.
- Si es una variable cuyo valor es **null**, entonces devolverá NULL.

Veamos el siguiente caso:

- Necesitamos identificar los diferentes tipos de datos que pueden darse en variables para un empleado de una empresa, el cual registrará sus nombres, sueldo, edad, fecha de nacimiento, y los países donde laboró:

```
<?php
    //Declaración y asignación de variable
    $nombre='Juan Perez';
    $sueldo=3050.78;
    $edad=35;
    $fechaNac='1979-10-10';
    $objTel=new stdClass();
    $países = array('Perú','Venezuela','Paraguay','Colombia');

    //Impresión de tipos de datos
    echo '$nombre es de tipo: '.gettype($nombre).'<br>';
    echo '$sueldo es de tipo: '.gettype($sueldo).'<br>';
    echo '$edad es de tipo: '.gettype($edad).'<br>';
    echo '$fechaNac es de tipo: '.gettype($fechaNac).'<br>';
    echo '$objTel es de tipo: '.gettype($objTel).'<br>';
    echo '$países es de tipo: '.gettype($países).'<br>';
?>
```

Debemos considerar que la fecha de nacimiento del empleado es considerado como tipo String, ya que en PHP no existe un tipo de datos para las fechas, como se muestra en la siguiente imagen:



Sólo fines educativos - FreeLibros

6.1.4 Función settype

Permite establecer un nuevo tipo de datos a una variable. Su formato es:

```
settype($variable,'Tipo_Datos');
```

Veamos los siguientes casos:

- En una empresa los empleados tienen un código asignado con el siguiente formato: 00005. Se necesita autogenerar un nuevo código cuando se registre un nuevo empleado, para esto se deberá aumentar en uno al código anterior:

```
<?php
    //Asignación de valores a variables
    $código = '00005';
    echo 'Código original: '.$código.'<br>';

    //Aplicamos settype
    settype($código, 'integer');
    $nuevoCodigo=sprintf("%05d",$codigo+1);

    //Impresión del nuevo código
    echo 'El valor de $código es: '.$nuevoCodigo;
?>
```

Resultado:

```
Código original: 00005
El valor de $codigo es: 00006
```

- Del caso anterior, generar los 10 códigos de los empleados de forma correlativa a partir del código 00005:

```
<?php
    //Asignación de valores a variables
    $código = '00005';
    echo 'Código original: '.$código.'<br>';

    //Aplicamos settype
    settype($código, 'integer');
    $nuevoCodigo=sprintf("%05d",$codigo+1);

    //Los siguientes 10 códigos autogenerados son:
    echo 'Los 10 códigos autogenerados son: <br>';
    for ($i=1;$i<=10;$i++){
        $nuevoCodigo=sprintf("%05d",$codigo+$i);
        echo $nuevoCodigo.'<br>';
    }
?>
```

Sólo fines educativos - FreeLibros

Resultado:

```
Código original: 00005
Los 10 códigos autogenerados son:
00006
00007
00008
00009
00010
00011
00012
00013
00014
00015
```

6.1.5 Función empty

Determina si una variable no tiene un valor asignado; además, para las últimas versiones de PHP se incorpora la comparación de expresiones vacías. Su formato es:

```
empty($variable)
```

Veamos algunos casos:

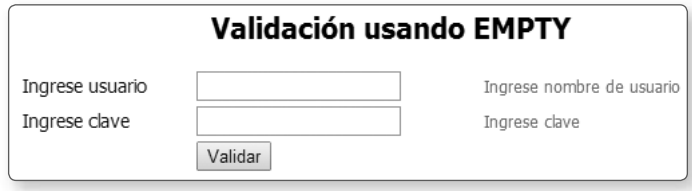
- Validando una variable de tipo **string**:

```
<?php
    $clave = '';
    if (empty($clave))
        echo 'La clave esta vacía';
?>
```

- Validando una variable de tipo entero:

```
<?php
    $numero = 0;
    if (empty($numero))
        echo 'El número esta vacío o es igual a cero';
?>
```

- Validando el usuario y la clave desde controles de formulario, hay que tener en cuenta que el control de la clave solo debe permitir el ingreso de 4 caracteres:



Sólo fines educativos - FreeLibros

```
<!DOCTYPE html>
<html>
    <head>
        <meta charset="UTF-8">
        <title>Validaciones</title>
        <link href="estilo.css" rel="stylesheet">
    </head>
    <body>
        <header>
            <h2 id="centrado">Validación usando EMPTY</h2>
        </header>
        <section>
        <?php
            error_reporting(0);
            //Capturando valores
            $usuario=$_POST['txtUsuario'];
            $clave=$_POST['txtClave'];

            //Validando el usuario y su clave
            $menUsuario='';
            $menClave='';
            if (empty($usuario))
                $menUsuario='Ingrese nombre de usuario';
            if (empty($clave))
                $menClave='Ingrese clave';
        ?>
        <form name="frmLogin" method="POST">
            <table border="0" width="550"
                   cellspacing="0" cellpadding="0">
                <tr>
                    <td>Ingrese usuario</td>
                    <td>
                     <input type="text" name="txtUsuario"
                         value="<?php echo $_POST['txtUsuario'];?>"/>
                    </td>
                    <td id="error"><?php echo $menUsuario; ?></td>
                </tr>
                <tr>
                    <td>Ingrese clave</td>
                    <td>
                     <input type="password" name="txtClave"
                          maxlength="4"
                          value="<?php echo $_POST['txtClave'] ?>"/>
                    </td>
                    <td id="error"><?php echo $menClave; ?></td>
                </tr>
                <tr>
                    <td></td>
                    <td><input type="submit" value="Validar" /></td>
                    <td></td>
                </tr>
            </table>
        </form>
        </section>
    </body>
</html>
```

Sólo fines educativos - FreeLibros

El archivo **estilo.css** contiene el siguiente código:

```
body{
    font-family: tahoma;
    font-size: 14px;
}

#centrado{
    text-align: center;
}
table {
    margin: auto;
}
#error{
    color: red;
    font-size: 12px;
}
```

6.1.6 Función is_integer

Determina si el valor de una variable es numérico entero. Su formato es:

```
is_integer($variable_numerica);
is_int($variable_numerica);
```

Veamos algunos casos:

- Mostrar el mensaje «Número ingresado correctamente» si el valor de una variable es numérica entera:

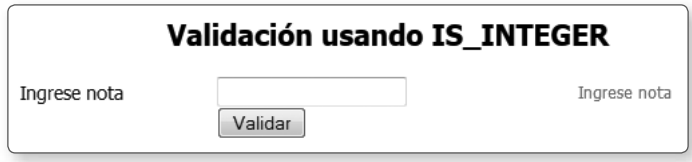
```
<?php
    $numero=123;
    if (is_integer($numero))
        echo 'Numero correcto';
    else
        echo 'Numero incorrecto';
?>
```

Posibles valores y respuestas del código:

$NUMERO	MENSAJE
123	'Número correcto'
123.25	'Número incorrecto'
'123'	'Número incorrecto'
'12A45'	'Número incorrecto'
-34	'Número correcto'

- Validar el ingreso de una nota en un determinado curso, para lo cual debe registrar un valor numérico correcto que corresponda a la nota de un alumno en un control TEXT; si está vacía, debe emitir un mensaje de «Ingrese nota»; si no es entero, emitir el mensaje «Nota incorrecta»; y si esta fuera del rango entre 0 y 20, emitir el mensaje «Rango de nota es de 0 a 20»:

Sólo fines educativos - FreeLibros




```
<!DOCTYPE html>
<html>
    <head>
        <meta charset="UTF-8">
        <title>Validaciones</title>
        <link href="estilo.css" rel="stylesheet">
    </head>
    <body>
        <header>
            <h2 id="centrado">Validación usando IS_INTEGER</h2>
        </header>
        <section>
        <?php
            error_reporting(0);

            //Capturando valores
            $nota=$_POST['txtNota']*1;

            //Validando
            $menNota='';
            if(empty($nota)) $menNota='Ingrese nota';
            if (!is_integer($nota))
                 $menNota='nota incorrecta';
                 elseif ($nota<0 || $nota>20)
                 $menNota='Rango de nota es de 0 a 20';
        ?>
        <form name="frmNota" method="POST">
            <table border="0" width="550"
                   cellspacing="0" cellpadding="0">
                <tr>
                    <td>Ingrese nota</td>
                    <td>
                      <input type="text" name="txtNota"
                          value="<?php echo $_POST['txtNota']; ?>"/>
                    </td>
                    <td id="error"><?php echo $menNota; ?></td>
                </tr>
                <tr>
                    <td></td>
                    <td><input type="submit" value="Validar" /></td>
                    <td></td>
                </tr>
            </table>
        </form>
        </section>
    </body>
</html>
```

Sólo fines educativos - FreeLibros

El archivo **estilo.css** contiene el siguiente código:

```
body{
    font-family: tahoma;
    font-size: 14px;
}

#centrado{
    text-align: center;
}
table {
    margin: auto;
}
#error{
    color: red;
    font-size: 12px;
}
```

6.1.7 Función is_double

Determina si el contenido de una variable numérica tiene un punto decimal para que sea considerado como tipo **double**. Su formato es:

```
is_double($variable);
```

Veamos algunos casos:

- Validar el registro de la altura de un paciente en su historial clínico, para lo cual se mostrará el mensaje «Altura registrada correctamente», si el valor es numérico decimal; caso contrario, emitir el mensaje «Altura ingresada de forma incorrecta..!!»:

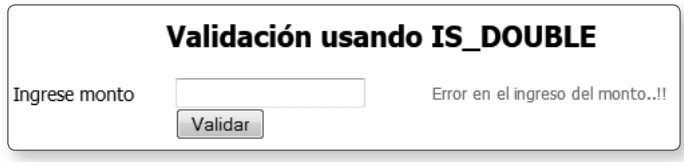
```
<?php
    $altura=1.72;
    if (is_double($altura))
        echo 'Altura registrada correctamente';
    else
        echo 'Altura ingresada de forma incorrecta..!!';
?>
```

Posibles valores y respuestas del código:

$ALTURA	MENSAJE
1.72	**"Altura registrada correctamente"**
172	**"Altura ingresada de forma incorrecta"**
'172cm'	**"Altura ingresada de forma incorrecta"**
'1.72'	**"Altura ingresada de forma incorrecta"**

- Validaremos el ingreso de un determinado monto de venta, este será registrado por un vendedor, el cual usará el control TEXT para dicho ingreso. Si el valor ingresado es número entero, negativo, está vacío o es un texto, emitir el mensaje «Error en el ingreso del monto..!!» :

Sólo fines educativos - FreeLibros




```
<!DOCTYPE html>
<html>
    <head>
        <meta charset="UTF-8">
        <title>Validaciones</title>
        <link href="estilo.css" rel="stylesheet">
    </head>
    <body>
        <header>
            <h2 id="centrado">Validación usando IS_DOUBLE</h2>
        </header>
        <section>
        <?php
            error_reporting(0);

            //Capturando valores
            $monto=$_POST['txtMonto']*1;

            //Validando
            $mensaje='';
            if (!is_double($monto) || !is_numeric($monto)
                  || empty($monto) || $monto<0)
                 $mensaje ='Error en el ingreso del monto..!!';
        ?>
        <form name="frmMonto" method="POST">
            <table border="0" width="550"
                   cellspacing="0" cellpadding="0">
                <tr>
                    <td>Ingrese monto</td>
                    <td>
                      <input type="text" name="txtMonto"
                          value="<?php echo $_POST['txtMonto']; ?>"/>
                    </td>
                    <td id="error"><?php echo $mensaje; ?></td>
                </tr>
                <tr>
                    <td></td>
                    <td><input type="submit" value="Validar" /></td>
                    <td></td>
                </tr>
            </table>
        </form>
        </section>
    </body>
</html>
```

Sólo fines educativos - FreeLibros

El archivo **estilo.css** contiene el siguiente código:

```
body{
    font-family: tahoma;
    font-size: 14px;
}

#centrado{
    text-align: center;
}
table {
    margin: auto;
}
#error{
    color: red;
    font-size: 12px;
}
```

6.1.8 Función is_string

Permite comprobar si el valor de una variable es de tipo cadena. Su formato es:

```
is_string($variable);
```

Veamos algunos casos:

- Validar el registro del código de un determinado empleado, para lo cual se mostrará el mensaje «Código registrado correctamente» si el valor es cadena; caso contrario, emitir el mensaje «Error en el ingreso de código»:

```
<?php
    $codigo='EMP001';
    if(is_string($codigo))
        echo 'Código registrado correctamente';
        else
        echo 'Error en el ingreso de código';
?>
```

Posibles valores y respuestas del código:

$CÓDIGO	MENSAJE
'EMP001'	“Código registrado correctamente”
1	“Error en el ingreso de código”
'001'	“Código registrado correctamente”
'@.com'	“Código registrado correctamente”
123.25	“Error en el ingreso de código”

Sólo fines educativos - FreeLibros

- Validaremos el ingreso para el nombre de un determinado empleado, este será registrado en un control TEXT y solo se permitirá el ingreso de cadena de caracteres en mayúsculas, minúsculas y con tildes. En caso se ingrese valores no textuales correctos, emitir el mensaje «Error en el nombre del empleado»:

```
<!DOCTYPE html>
<html>
    <head>
        <meta charset="UTF-8">
        <title>Validaciones</title>
        <link href="estilo.css" rel="stylesheet">
    </head>
    <body>
        <header>
            <h2 id="centrado">Validación usando IS_STRING</h2>
        </header>
        <section>
        <?php
            error_reporting(0);

            //Capturando valores
            $empleado=$_POST['txtEmpleado'];

            //Validando
            $mensaje='';
            $permitidos = '/^[A-Z üÜáéíóúÁÉÍÓÚñÑ]{1,100}$/i';
            if (!preg_match($permitidos,$empleado)
                                        || !is_string($empleado))
                $mensaje ='Error en el nombre del empleado..!!';
        ?>
        <form name="frmEmpleado" method="POST">
            <table border="0" width="750"
                   cellspacing="0" cellpadding="0">
                <tr>
                    <td>Ingrese nombre del empleado</td>
                    <td>
                     <input type="text" name="txtEmpleado"
                       value="<?php echo $_POST['txtEmpleado']; ?>"/>
                    </td>
                    <td id="error"><?php echo $mensaje; ?></td>
                </tr>
                <tr>
                    <td></td>
                    <td><input type="submit" value="Validar" /></td>
                    <td></td>
                </tr>
            </table>
        </form>
        </section>
    </body>
</html>
```

La función **preg_match** permite encontrar coincidencias de una expresión en una determinada cadena de caracteres. Ampliaremos esta función en el tema «funciones de cadena».

Sólo fines educativos - FreeLibros

6.1.9 Función var_dump

Permite informar sobre el tipo y el valor de una determinada variable o expresión. Su formato es:

```
var_dump($variable);
```

Veamos el siguiente caso:

- Se debe mostrar información acerca de los valores y variables que puede tener un determinado alumno:

```
<?php
    //Ingreso de valores
    $empleado = 'Ángela Torres Lázaro';
    $fechaNac = '16/05/2005';
    $talla = 1.30;
    $edad = 9;

    //Impresión
    var_dump($empleado);
    var_dump($fechaNac);
    var_dump($talla);
    var_dump($edad);
?>
```

Resultado:

```
string 'Angela Torres Lázaro' (length=21)

string '16/05/2005' (length=10)

float 1.3

int 9
```

Posibles valores y respuestas del código:

VAR_DUMP	IMPRESIÓN
$empleado	String 'Ángela Torres Lázaro' (Longitud=21)
$fechaNac	String '16/05/2005' (Longitud=10)
$talla	Float 1.3
$edad	Int 9

6.2 FUNCIONES DE CADENA

Antes de comenzar con la especificación de las funciones que puede tener una variable string, debemos recordar que un string admite todo tipo de caracteres, inclusive numéricos. Desde aquí mencionaremos que, para especificar un carácter, necesitamos cualquiera de los siguientes formatos:

- Comillas simples: Se pueden presentar los siguientes casos:

```
$código='EMP001';
$edad='24';
$email='ftorres@gmail.com';
$fechaNac='2005-05-16';
```

Sólo fines educativos - FreeLibros

- Comillas dobles: Se pueden presentar los siguientes casos:

```
$código="EMP001";
$edad="24";
$email="ftorres@gmail.com";
$fechaNac="2005-05-16";
```

Veamos el siguiente caso, si se necesita imprimir el valor asignado a una variable **sueldo** usando la comilla simple, sería de la siguiente manera:

```
<?php
    $sueldo = 1000;
    echo 'El sueldo es:'.$sueldo;
?>
```

Notemos que para imprimir la variable sueldo se necesita concatenar con el operador **punto**. En cambio, si usamos la comilla doble el código cambiaría a:

```
<?php
    $sueldo = 1000;
    echo "El sueldo es: $sueldo";
?>
```

6.2.1 Función strlen

Permite determinar la longitud de caracteres contenidos en el valor de una variable de tipo string. Su formato es:

```
$variable=strlen($variable_string);
```

Veamos algunos casos:

- Validar el ingreso de la edad de una persona. Solo se debe permitir el ingreso de 2 caracteres para la edad, que no sea inferior a 18 ni superior a 65. Mostrar un mensaje por cada evaluación:

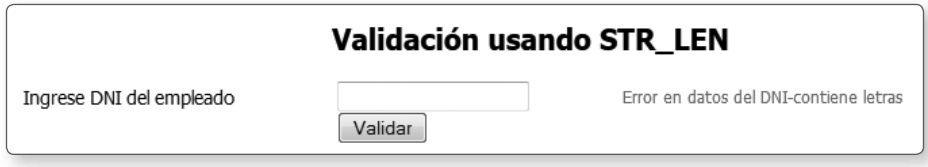
```
<?php
    $edad = 18;

    if (strlen($edad)!=2)
        $mensaje='La edad debe tener dos dígitos';
        elseif ($edad<18)
            $mensaje='La edad debe superior o igual a 18';
        elseif ($edad>65)
            $mensaje='La edad no debe ser superior a 65';
        else
            $mensaje='La edad ingresada es correcta..!!';
    echo $mensaje;
?>
```

Sólo fines educativos - FreeLibros

Posibles valores y respuestas del código:

$EDAD	MENSAJE
18	'La edad ingresada es correcta..!!'
16	'La edad debe superior o igual a 18'
66	'La edad no debe ser superior a 65'
7	'La edad debe tener dos dígitos'

- Validaremos el ingreso de un número de DNI perteneciente a un empleado; en la cual no debe permitir caracteres tipo letras y la longitud exactamente 8, en caso de errores mostrar los mensajes convenientes.




```
<!DOCTYPE html>
<html>
    <head>
        <meta charset="UTF-8">
        <title>Validaciones</title>
        <link href="estilo.css" rel="stylesheet">
    </head>
    <body>
        <header>
            <h2 id="centrado">Validación usando STR_LEN</h2>
        </header>
        <section>
        <?php
            error_reporting(0);

            //Capturando valores
            $dni=$_POST['txtDNI'];

            //Validando
            $mensaje='';
            if(!preg_match('/^[[:digit:]]+$/', $dni))
                $mensaje='Error en datos del DNI-contiene letras';
                elseif (strlen($dni)!=8)
                $mensaje='DNI debe tener 8 caracteres numéricos';
        ?>
        <form name="frmEmpleado" method="POST">
            <table border="0" width="750"
                   cellspacing="0" cellpadding="0">
                <tr>
                    <td>Ingrese DNI del empleado</td>
                    <td>
                      <input type="text" name="txtDNI"
                          value="<?php echo $_POST['txtDNI']; ?>"/>
                    </td>
                    <td id="error"><?php echo $mensaje; ?></td>
                </tr>
                <tr>
                    <td></td>
                    <td><input type="submit" value="Validar" /></td>
```

Sólo fines educativos - FreeLibros

```
                    <td></td>
                </tr>
            </table>
        </form>
        </section>
    </body>
</html>
```

El archivo **estilo.css** contiene el siguiente código:

```
body{
    font-family: tahoma;
    font-size: 14px;
}
#centrado{
    text-align: center;
}
#error{
    color: red;
    font-size: 12px;
}
```

6.2.2 Función strpos

Determina la posición del primer valor encontrado en una cadena de caracteres; hay que tener en cuenta que el punto de inicio de las posiciones es cero. Su formato es:

```
$variable_posicion = strpos($variable_a_evaluar,'contenido');
```

Se presentan algunas variaciones de la función:

VARIACIÓN	DESCRIPCIÓN
strrpos($variable_a_evaluar,'contenido')	Devuelve la posición de la última vez que se encontró un valor dentro de una cadena.
stripos($variable_a_evaluar,'contenido')	Tiene la misma función que **strpos**, pero adiciona que no distingue entre mayúsculas y minúsculas.
strripos($variable_a_evaluar,'contenido')	Devuelve la posición de la última vez que se encontró un valor dentro de una cadena, sin distinguir las mayúsculas de las minúsculas.

Veamos el siguiente caso:

- Desde una dirección de correo electrónico determinar en qué posición se encuentra el carácter @:

```
<?php
    $email = 'manuel.torresr@hotmail.com';
    $pos=strpos($email,'@')+1;
    echo '@ se encuentra en la posición'.$pos;
?>
```

Sólo fines educativos - FreeLibros

Resultado:

Notemos que a la variable **pos** se le aumenta un valor, ya que las posiciones empiezan el cero.

6.2.3 Función strcmp

Permite hacer una comparación binaria entre dos valores, su formato es:

```
$variable_resultante = strcmp($variable1,$variable2);
```

El resultado de la expresión **strcmp** dependerá de las variables a comparar, por tanto, podríamos decir que:

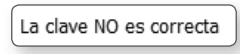
- Si la **variable1** es menor a la **variable2** entonces devolverá -1.
- Si la **variable1** es mayor a la **variable2** entonces devolverá 1.
- Si la **variable1** es exactamente igual a la **variable2** entonces devolverá 0.

Veamos el siguiente caso:

- Determinar si la cadena de caracteres que representan la clave son iguales o no; mostrar los mensajes correspondientes.

```
<?php
    $clave = 'Sócrates';
    $iClave = 'socrateS';

    if (strcmp($clave, $iClave)==0)
            echo 'La clave es correcta';
            else
            echo 'La clave NO es correcta';
?>
```

Resultado:



6.2.4 Función strstr

Permite obtener los caracteres anteriores o posteriores al encuentro de un valor en una cadena de caracteres; hay que tener en cuenta que la función **strstr** distingue entre mayúsculas y minúsculas. Su formato es:

```
$variable_obtenido = strstr($variable,'valor',true);
$variable_obtenido = strstr($variable,'valor',false);
```

La opción **true** permite obtener los caracteres, desde la posición inicial hasta la posición donde encontró el carácter. Mientras que **false** obtiene los caracteres desde la posición encontrada hasta la posición final de la cadena.

Sólo fines educativos - FreeLibros

Veamos el siguiente caso:

- Desde una dirección de correo electrónico imprimiremos el nombre del servidor y el nombre del usuario:

```
<?php
    $email  = 'manuel.torresr@hotmail.com';
    $servidor = strstr($email, '@');

    $usuario = strstr($email, '@', true);
    echo 'El dominio del correo electrónico es: '.$servidor.'<br>';
    echo 'El usuario del correo electrónico es: '.$usuario;
?>
```

Resultado:

```
El dominio del correo electrónico es: @hotmail.com
El usuario del correo electrónico es: manuel.torresr
```

6.2.5 Función substr

Permite devolver una parte de una cadena de caracteres, especificando la posición inicial y cuantos caracteres desea obtener. Su formato es:

```
$variable_obtenida = substr($variable,posicion_inicial, cantidad_caracteres);
```

Los valores devueltos serán obtenidos desde **$variable**, mientras que la posición inicial es el punto de inicio de la captura de valores, y **cantidad de caracteres** define el total de caracteres a obtener a partir de la posición indicada; esta última es opcional pues sino lo indica se obtendrán todos los valores a partir de la posición indicada.

Veamos el siguiente caso:

- Desde una dirección de correo electrónico imprimiremos el nombre del servidor y el nombre del usuario:

```
<?php
    $email  = 'manuel.torresr@hotmail.com';
    $servidor = substr($email, strpos($email,'@'));

    $usuario = substr($email, 0, strpos($email,'@'));
    echo 'El dominio del correo electrónico es: '.$servidor.'<br>';
    echo 'El usuario del correo electrónico es: '.$usuario;
?>
```

Resultado:

```
El dominio del correo electrónico es: @hotmail.com
El usuario del correo electrónico es: manuel.torresr
```

Sólo fines educativos - FreeLibros

6.2.6 Funciones ltrim, rtrim, chop y trim

Presentamos las siguientes descripciones:

FUNCIÓN	DESCRIPCIÓN
ltrim	La función **ltrim** permite eliminar espacios en blanco, tabulaciones o cambios de línea desde el lado izquierdo de una cadena de caracteres.
rtrim	La función **rtrim** permite eliminar espacios en blanco, tabulaciones o cambios de línea desde el lado derecho de una cadena de caracteres.
chop	Elimina caracteres desde el lado derecho de una cadena de caracteres, tiene analogía con **rtrim** siempre y cuando no se especifique parámetros a la función **chop**.
trim	La función **trim** permite eliminar espacios en blanco, tabulaciones o cambios de línea en ambos lados de una cadena de caracteres.

Veamos el siguiente caso:

- Veamos el uso de las funciones en diferentes cadenas:

```
<?php
//Asignación de valores
$cadena1 = 'Fundamentos de Programación con PHP';
$cadena2 = '                Fundamentos de Programación con PHP';
$cadena3 = 'Fundamentos de Programación con PHP                ';
$cadena4 = '         Fundamentos de Programación con PHP       ';

//Impresión
echo chop($cadena1,'Programación con PHP').'<br>';
echo ltrim($cadena2).'<br>';
echo rtrim($cadena3).'<br>';
echo trim($cadena4).'<br>';
?>
```

Resultado:

```
Fundamentos de
Fundamentos de Programación con PHP
Fundamentos de Programación con PHP
Fundamentos de Programación con PHP
```

6.2.7 Función str_replace

Permite reemplazar una cadena por otra; tantas veces sea encontrado. Su formato es:

```
$variable_obtenida=str_replace($cadena);
```

Veamos el siguiente caso:

- A partir del código de un alumno con el formato 2014CI0001, debemos modificarlo por 2014ET0001 por un error de registro en la base de datos:

Sólo fines educativos - FreeLibros

```php
<?php
    //Asignación de valores
    $código = '2014CI0001';

    //Impresion
    echo str_replace('CI','ET', $codigo); //Resp: 2014ET0001
?>
```

Resultado:

```
2014ET0001
```

6.2.8 Funciones strtolower y strtoupper

Son funciones que tienen el control de cambios de mayúsculas a minúsculas (strtolower) y de minúsculas a mayúsculas (strtoupper) de una cadena de caracteres. Su formato es:

```php
$variable_minusculas = strtolower($variable);
$variable_mayusculas = strtoupper($variable);
```

Veamos el siguiente caso:

- El encargado del área de personal de una empresa necesita registrar los datos de un determinado empleado, a quien se le solicita que ingrese sus datos por medio de una página web ofrecida por la empresa. A partir de dicha información, asegurar que el código del empleado sea en mayúsculas y que el correo electrónico sea estrictamente en minúsculas:

```php
<?php
    //Asignación de valores
    $código = 'emp001_14a';
    $email = 'MANUEL.TORRESR@HOTMAIL.COM';

    //Impresión
    echo 'El código registrado es: '.strtoupper($código).'<br>';
    echo 'El correo electrónico registrado es: '.strtolower($email);
?>
```

Resultado:

```
El código registrado es: EMP001_14A
El correo electrónico registrado es: manuel.torresr@hotmail.com
```

6.2.9 Función preg_match

Permite realizar una comparación entre dos cadenas de caracteres. Su formato es:

```php
preg_match('/patron/',$variable);
```

Sólo fines educativos - FreeLibros

Para iniciar un patrón debemos colocar los símbolos / patrón, a continuación describiremos algunos formatos:

Símbolo	Descripción
.	El punto representa a cualquier carácter alfanumérico dentro de una cadena de caracteres.
^	Indica el carácter de inicio dentro de una cadena caracteres. Por ejemplo, **^a** indica que la cadena inicia con la letra **a**.
$	Indica con qué carácter termina dentro de una cadena de caracteres. Por ejemplo, **$a** indica que la cadena termina con la letra **a**.
+	Indica que el carácter se puede repetir una o más veces. Por ejemplo, **a+** indica que **a** se repetirá una o más veces dentro de la cadena de caracteres.
*	Indica que el carácter se puede repetir de cero veces a más. Por ejemplo, **a*** indica que **a** podría estar en la cadena o podría repetirse muchas veces.
?	Indica que el carácter se puede repetir cero o, por lo menos, una vez. Por ejemplo, **a?** indica que **a** se puede encontrar en la cadena de caracteres solo una vez o nunca.
\	Sirve para concatenar el formato con símbolos como ^ [ ] . ( ) | * ? { } \.
i	Indica que los caracteres no tienen distinción entre mayúsculas o minúsculas.
( )	Indica que los caracteres especificados dentro se buscarán de forma agrupada dentro de la cadena de caracteres. Por ejemplo (**gma**) indica que se buscarán los caracteres agrupados **gma** dentro de la cadena, también se puede usar **\b**.
|	Indica que los caracteres separados por **|** se buscarán independientemente; podría encontrarse uno o los dos a la vez dentro de una cadena de caracteres. Por ejemplo, **a|b|c** indica que se puede encontrar **a** o **b** o **c** dentro de la cadena.
{n}	Indica la cantidad de veces que se puede repetir un determinado carácter. Por ejemplo **a{3}** indica que a se repetirá tres veces dentro de una cadena de caracteres.
{n,m}	Indica la cantidad de veces, por rango, que se puede repetir un determinado carácter. Por ejemplo, **a{1,5}** indica que **a** se repetirá de una a cinco veces dentro de una cadena de caracteres.
[ ]	Indica un patrón de búsqueda de caracteres, combinando los símbolos anteriores. Veamos algunos ejemplos: • [a-z] Representa a todas las letras en minúsculas, también puede usar \D. • [A-Z] Representa a todas las letras en mayúsculas, también puede usar \D. • [0-9] Representa a todos los valores numéricos enteros, también puede usar \d. • [[:alnum:]] Representa a todos los caracteres alfanuméricos. • [[:alpha:]] Representa exclusivamente a caracteres. • [[:digit:]] Representa exclusivamente a números enteros. • [[:lower:]] Representa a cualquier carácter en minúscula. • [[:upper:]] Representa a cualquier carácter en mayúscula. • [[:space:]] Representa un espacio en blanco, también puede usar \s.

Veamos los siguientes casos:

- El código de un empleado está compuesto por tres letras iniciales y el número de empleado de forma correlativa; se necesita verificar que los caracteres **emp** o EMP se encuentren dentro del código registrado:

```
<?php
    $codigo='emp001';
    if (preg_match('/EMP/i',$codigo))
        echo "Caracteres encontrados correctamente";
        else
        echo "Caracteres NO encontrados..!!";
?>
```

Sólo fines educativos - FreeLibros

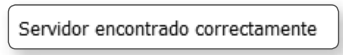
Resultado:

Considere que el atributo **i** en el patrón permite omitir la distinción entre mayúsculas y minúsculas; por lo tanto, será igual buscar EMP que **emp**.

- Desde el correo electrónico de un empleado se necesita determinar si pertenece al servidor Hotmail por medio de un mensaje:

```
<?php
    $email='manuel.torresr@hotmail.com';
     $buscar='hotmail';
    if (preg_match("/\b$buscar\b/i",$email))
        echo " Servidor encontrado correctamente";
        else
        echo " Servidor NO encontrado..!!";
?>
```

Resultado:

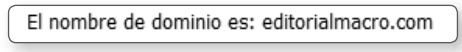


Considere que el atributo **\b** permite buscar la palabra completa dentro de una cadena de caracteres.

- Desde una dirección URL imprimiremos el nombre del servidor:

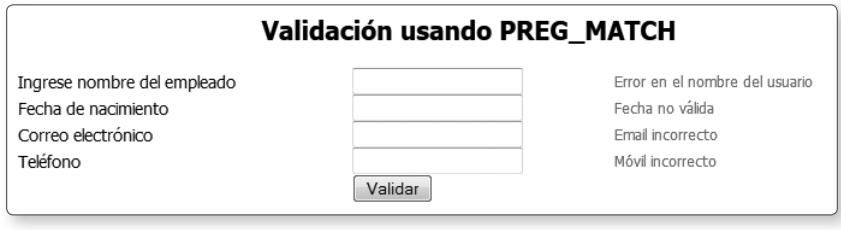
```
<?php
//Asignando valores
$url='http://www.editorialmacro.com/desarrolladores.html';
preg_match('@^(?:http://)?([^/]+)@i',$url, $coincidencias);
$servidor = $coincidencias[1];

// obtiene el nombre exacto del servidor
preg_match('/[^.]+\.[^.]+$/', $servidor, $coincidencias);
echo "El nombre de dominio es: {$coincidencias[0]}\n";
?>
```

Resultado:



- Validar el registro de datos de un empleado, para lo cual ingresará su nombre completo, fecha de nacimiento, correo electrónico y teléfono. Tener en cuenta:
  - En el nombre del empleado solo estará permitido el ingreso de letras y espacios.
  - En la fecha de nacimiento solo se permitirá el ingreso de fechas con formato válido, como dd/mm/aaaa.
  - El correo electrónico deberá tener el formato adecuado: usuario@servidor.com.
  - El teléfono contará con el formato 999999999, el cual debe iniciar obligatoriamente en 9 y contar con 9 dígitos.

Sólo fines educativos - FreeLibros




```
<!DOCTYPE html>
<html>
    <head>
        <meta charset="UTF-8">
        <title>Validaciones</title>
        <link href="estilo.css" rel="stylesheet">
    </head>
    <body>
        <header>
            <h2 id="centrado">Validación usando PREG_MATCH</h2>
        </header>
        <section>
        <?php
            error_reporting(0);

            //Capturando valores
            $nombre=$_POST['txtNombre'];
            $fecha=$_POST['txtFecha'];
            $email=$_POST['txtEmail'];
            $fono=$_POST['txtFono'];

            //Validando
            $mNombre='';
            if (!preg_match('/^[A-Z üÜáéíóúÁÉÍÓÚñÑ]{1,100}$/i',
                                                    $nombre))
                    $mNombre='Error en el nombre del usuario';

            $mFecha='';
            if (!preg_match('/^\d{1,2}\/\d{1,2}\/\d{4}$/', $fecha))
                    $mFecha='Fecha no valida';

            $mEmail='';
            if (!preg_match('/^([a-zA-Z0-9])+([a-zA-Z0-9\._-])*
                 @([a-zA-Z0-9_-])+([a-zA-Z0-9\._-]+)+$/', $email))
                    $mEmail='Email incorrecto';

            $mFono='';
            if (!preg_match('/^[9][0-9]{8}$/', $fono))
                    $mFono='Movil incorrecto';
        ?>

        <form name="frmEmpleado" method="POST">
            <table border="0" width="750"
                   cellspacing="0" cellpadding="0">
             <tr>
              <td>Ingrese nombre del empleado</td>
              <td>
                <input type="text" name="txtNombre"
                    value="<?php echo $_POST['txtNombre']; ?>"/>
              </td>
              <td id="error"><?php echo $mNombre; ?></td>
```

Sólo fines educativos - FreeLibros

```
                </tr>
                <tr>
                 <td>Fecha de nacimiento</td>
                 <td><input type="text" name="txtFecha"
                      value="<?php echo $_POST['txtFecha']; ?>" /></td>
                 <td id="error"><?php echo $mFecha; ?></td>
                </tr>
                <tr>
                  <td>Correo electronico</td>
                  <td><input type="text" name="txtEmail"
                        value="<?php echo $_POST['txtEmail']; ?>" />
                  </td>
                  <td id="error"><?php echo $mEmail; ?></td>
                </tr>
                <tr>
                  <td>Teléfono</td>
                  <td><input type="text" name="txtFono"
                        value="<?php echo $_POST['txtFono']; ?>" /></td>
                  <td id="error"><?php echo $mFono; ?></td>
                </tr>
                <tr>
                  <td></td>
                  <td><input type="submit" value="Validar" /></td>
                  <td></td>
                </tr>
              </table>
          </form>
          </section>
      </body>
  </html>
```

6.2.10 Función explode

Permite dividir una cadena en pequeñas porciones de cadenas de caracteres, siempre que se siga un criterio para la división. Su formato es:

```
$variable_arreglo=explode('Criterio',$variable_a_descomponer);
```

La **variable_arreglo** recibirá la información descompuesta de acuerdo al criterio, y para poder mostrarlo al usuario deberá acceder como si fuera arreglo unidimensional, es decir:

```
$arreglo_variable[0]; muestra el primero bloque.
$arreglo_variable[1]; muestra el segundo bloque.
$arreglo_variable[…]; ...
$arreglo_variable[n-1]; muestra el último bloque.
```

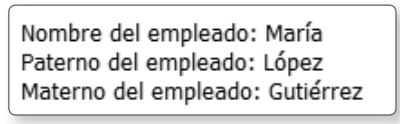
Veamos algunos casos:

- Dividir el nombre completo de un empleado en nombres, apellido paterno y apellido materno; hay que tener en cuenta que solo se debe registrar un nombre, un apellido paterno y un materno:

```
<?php
    $empleado='María López Gutiérrez';

    $nombre=explode(' ',$empleado);
    echo 'Nombre del empleado: '.$nombre[0].'<br>';
    echo 'Paterno del empleado: '.$nombre[1].'<br>';
    echo 'Materno del empleado: '.$nombre[2].'<br>';
?>
```

Sólo fines educativos - FreeLibros

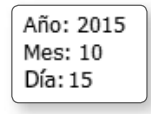
Resultado:



- Mostrar el día, mes y año a partir de una fecha completa:

```
<?php
    $hoy='2015-10-15';

    $fecha=explode('-',$hoy);
    echo 'Año: '.$fecha[0].'<br>';   //Resp: 2015
    echo 'Mes: '.$fecha[1].'<br>';   //Resp: 10
    echo 'día: '.$fecha[2];          //Resp: 15
?>
```

Resultado:



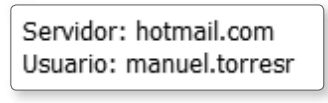
Otra forma de expresar el mismo caso podría ser:

```
<?php
    $hoy='2015-10-15';

    list($año,$mes,$dia)=explode('-',$hoy);
    echo 'Año: '.$año.'<br>';                //Resp: 2015
    echo 'Mes: '.$mes.'<br>';                //Resp: 10
    echo 'día: '.$día;                       //Resp: 15
?>
```

- A partir de un correo electrónico válido, obtener el nombre del usuario y el servidor de correo.

```
<?php
    $email='manuel.torresr@hotmail.com';

    $correo=explode('@',$email);
    echo 'Servidor: '.$correo[1].'<br>';     //Resp: hotmail.com
    echo 'Usuario: '.$correo[0];                      //Resp:manuel.torresr
?>
```

Resultado:



Sólo fines educativos - FreeLibros

6.2.11 Función strrev

Permite invertir los caracteres contenidos en una variable de tipo cadena. Su formato es:

```
$variable_invertida=strrev($variable);
```

Veamos el siguiente caso:

- A partir de un código de empleado imprimir la cadena de revés:

```
<?php
    $código='100PME';
    echo 'Nuevo código es: '.strrev($código); //Resp: EMP001
?>
```

Resultado:

```
Nuevo código es: EMP001
```

6.2.12 Función str_repeat

Permite devolver un número determinado de veces una cadena de caracteres. Su formato es:

```
str_repeat('valor_repetido',veces_repetidas);
```

Veamos el siguiente caso:

- A partir de un número entero mostrar el siguiente formato 000001:

```
<?php
    $n=1;
    echo 'El numero de factura es: '.  str_repeat('0',5).$n;
?>
```

Resultado:

```
El número de factura es: 000001
```

6.2.13 Función str_pad

Permite rellenar una cadena en otra un número determinado de veces. Su formato es:

```
str_pad($variable, numero_repeticiones, 'carácter', tipo);
```

Donde:

- **$variable**: Es el valor base para el relleno de caracteres, hay que tener en cuenta que el relleno puede darse a la izquierda o derecha de este valor.
- **Numero_repeticiones**: Es el valor numérico que indica cuántas veces se repetirá el carácter en la cadena.
- **Carácter**: Es el valor a repetirse.

Sólo fines educativos - FreeLibros

- **Tipo**: Es la definición opcional, un tipo determina la posición del valor repetido. Se presentan los siguientes tipos:
  - STR_PAD_RIGHT: Rellena por el lado derecho.
  - STR_PAD_LEFT: Rellena por el lado izquierdo.
  - STR_PAD_BOTH: Rellena por la derecha y la izquierda a la vez.

Cuando no se especifica un tipo se asume el valor STR_PAD_RIGHT.

Veamos el siguiente caso:

- A partir de un número entero mostrar el siguiente formato 000001:

```
<?php
    $n = 1;
    echo str_pad($n,5,'0',STR_PAD_LEFT);     //Resp:00001
?>
```

6.3 Funciones numéricas

Las funciones numéricas para PHP están preparadas para valores que estén dentro de los límites permitidos por los tipos de datos **integer** y **float**. Hay que tener en cuenta que un valor numérico es asignado a una variable sin especificar comillas ni otra clase de símbolo, entonces podríamos mencionar las siguientes asignaciones:

- Asignación de números enteros:

```
$n=10;
$edad=24;
```

- Asignación de números decimales o fraccionarios:

```
$sueldo=1250.50;
$talla=1.75;
```

- Forzar una cadena numérica a número entero o **float**:

```
$n='10' * 1;
$sueldo='1250.50' * 1.0;
```

6.3.1 Función abs

Permite devolver el mismo valor asignado en forma positiva, considere que **abs** no modifica los tipos de datos originales del valor numérico. Su formato es:

```
abs($variable_numerica);
```

Sólo fines educativos - FreeLibros

Veamos el siguiente caso:

- A partir de una expresión numérica que puede resultar positiva o negativa se le aplicará la función **abs** para imprimir un número entero:

```
<?php
    $monto=1500;
    $descuento=10/100.0 * $monto;
    $montoDescontado=($descuento-$monto);
    echo 'El nuevo monto es: '.abs($montoDescontado); //Resp. 1350
?>
```

6.3.2 Función ceil

Permite redondear un valor numérico al valor entero superior. Su formato es:

```
ceil($variable);
```

Veamos el siguiente caso:

- A partir de un promedio imprimir el valor numérico superior:

```
<?php
    $promedio=10.3;
    echo 'El nuevo monto es: '.ceil($promedio); //Resp. 11
?>
```

6.3.3 Función exp

Permite devolver un valor exponencial de **e**, el cual tiene el valor 2.7182. Su formato es:

```
exp($variable_numerica);
```

Veamos el siguiente caso:

- A partir de un número entero determinar el valor exponencial:

```
<?php
    $n=1;
    echo 'El nuevo monto es: '.exp($n); //Resp. 2.7182
?>
```

6.3.4 Función floor

Permite redondear un valor numérico al valor entero inferior. Su formato es:

```
floor($variable_numerica);
```

Sólo fines educativos - FreeLibros

Veamos el siguiente caso:

- A partir de un valor numérico redondearlo al valor entero inferior:

```
<?php
    $n=4.3;
    echo 'El numero redondeado es:'. floor($n); //Resp. 4
?>
```

6.3.5 Función getrandmax

Permite mostrar el número máximo entero de los valores aleatorios permitidos por PHP. Su formato es:

```
getrandmax();
```

Veamos el siguiente caso:

- Mostrar el máximo valor numérico aleatorio permitido por PHP:

```
<?php
    echo 'El máximo valor aleatorio es: '.getrandmax(); //Resp. 32767
?>
```

6.3.6 Función max

Permite devolver el máximo valor numérico de un conjunto de valores del mismo tipo. Su formato es:

```
max($variable_coleccion);
```

Veamos el siguiente caso:

- A partir de un conjunto de números enteros almacenados en un arreglo llamado **$números**, determinar el mayor elemento:

```
<?php
    $números=array(1,4,26,7);
    echo 'El elemento mayor es: '.max($números);  //Resp: 26
?>
```

6.3.7 Función min

Permite devolver el mínimo valor numérico registrado en un conjunto de valores del mismo tipo. Su formato es:

```
min($variable_coleccion);
```

Sólo fines educativos - FreeLibros

Veamos el siguiente caso:

- A partir de un conjunto de números enteros almacenados en un arreglo llamado $números, determinar el menor elemento:

```
<?php
    $números=array(1,4,26,7);
    echo 'El menor elemento es: '.min($números); //Resp: 1
?>
```

6.3.8 Función mt_rand

Permite generar un número entero aleatorio a partir de un valor inicial y final:

```
mt_rand($valor_inicial,$valor_final);
```

Veamos el siguiente caso:

- Generar un número aleatorio única y exclusivamente de tres cifras:

```
<?php
    $valor_inicial=100;
    $valor_final=999;

    echo mt_rand($valor_inicial, $valor_final);
?>
```

Una vez ejecutada la aplicación puede generar un nuevo número aleatorio presionando F5 en su navegador.

6.3.9 Función pi

Permite devolver el valor fraccionario de PI, es decir 3.1415926535898. Su formato es:

```
pi();
M_PI;
```

Veamos el siguiente caso:

- Imprimir el valor PI:

```
<?php
    echo pi(); // Resp: 3.1415926535898
    echo M_PI; // Resp: 3.1415926535898
?>
```

6.3.10 Función pow

Permite elevar un valor numérico a la N potencia. Su formato es:

```
$variable=pow($valor_base,valor_potencia);
```

Sólo fines educativos - FreeLibros

Veamos el siguiente caso:

- A partir del número entero 2, determinar el cuadrado, cubo, raíz cuadrada y raíz cúbica:

```
<?php
    $n=2;
    $cuadrado=pow($n,2);
    $cubo=pow($n,3);
    $rcuadrada=pow($n,1.0/2.0);
    $rcubica=pow($n,1.0/3.0);

    echo 'El valor ' .$n.' al cuadrado es: '.$cuadrado; // Resp: 4
    echo '<br>El valor ' .$n.' al cubo es: '.$cubo; // Resp: 8
    echo '<br>La raíz cuadrada de ' .$n.' es: '.$rcuadrada; // Resp: 1.4142
    echo '<br>La raíz cubica de ' .$n.' es: '.$rcubica; // Resp: 1.2599
?>
```

6.3.11 Función round

Permite devolver un número redondeado a N valores decimales. Su formato es:

```
round($variable_decimal,numero_decimales);
```

Veamos el siguiente caso:

- A partir de un número decimal, mostrar el valor redondeado a 0, 1 y 2 decimales:

```
<?php
    $nota=10.5;
    echo 'La nota es: '.round($nota);  //Resp: 11
    echo 'La nota es: '.round($nota,1);  //Resp: 10.5
    echo 'La nota es: '.round($nota,2);  //Resp: 10.5
?>
```

6.3.12 Función sqrt

Permite devolver la raíz cuadrada de un número. Su formato es:

```
sqrt($variable);
```

Veamos el siguiente caso:

- Muestre la raíz cuadrada de un número:

```
<?php
    $n=2;
    echo 'La raíz cuadrada de '.$n .' es: '.sqrt($n);
?>
```

Sólo fines educativos - FreeLibros

Resultado:

```
La raíz cuadrada de 2 es: 1.4142135623731
```

6.4 FUNCIONES DE FECHA Y HORA

PHP considera a las fechas como una cadena de caracteres, haciendo que sus funciones detecten una fecha válida y, a partir de allí, obtener distintos valores.

6.4.1 Función date

Permite retornar valores referentes a la fecha y hora actual. Su formato es:

```
date(formato);
```

Veamos los siguientes casos:

- Mostrar la fecha actual:

```
<?php
    echo 'Fecha actual: '.date('d-m-y');
?>
```

Resultado:

```
Fecha actual: 02-08-14
```

- Listaremos los formatos de la función date.

```
<?php
    echo 'AM o PM: '.date('A');
    echo '<br>am o pm: '.date('a');
    echo '<br>Día actual de dos cifras: '.date('d');
    echo '<br>Día actual sin ceros a la izquierda: '.date('j');
    echo '<br>Mes actual: '.date('F');
    echo '<br>Mes actual 3 letras: '.date('M');
    echo '<br>Número de mes de dos cifras: '.date('m');
    echo '<br>Número de mes sin ceros a la izquierda: '.date('n');
    echo '<br>Año de 4 cifras: '.date('Y');
    echo '<br>Año de 2 cifras: '.date('y');
    echo '<br>Hora actual 0-23: '.date('G');
    echo '<br>Hora actual 0-23 de dos cifras: '.date('H');
    echo '<br>Hora actual 1-12: '.date('g');
    echo '<br>Hora actual 1-12 de dos cifras: '.date('h');
    echo '<br>Minuto actual de dos cifras: '.date('i');
    echo '<br>Segundos actual de dos cifras: '.date('s');
    echo '<br>Número de días del mes actual: '.date('t');
?>
```

Sólo fines educativos - FreeLibros

Resultado:

```
AM o PM: PM
am o pm: pm
Día actual de dos cifras: 01
Día actual sin ceros a la izquierda: 1
Mes actual: August
Mes actual 3 letras: Aug
Número de mes de dos cifras: 08
Número de mes sin ceros a la izquierda: 8
Año de 4 cifras: 2014
Año de 2 cifras: 14
Hora actual 0-23: 17
Hora actual 0-23 de dos cifras: 17
Hora actual 1-12: 5
Hora actual 1-12 de dos cifras: 05
Minuto actual de dos cifras: 44
Segundos actual de dos cifras: 54
Número de días del mes actual: 31
```

- Mostrar la fecha actual en el siguiente formato: Lunes 30th de Junio 2015 19:13:20 PM.

```
<?php
    echo date('l jS \of F Y h:i:s A');
?>
```

Resultado:

```
Friday 1st of August 2014 05:45:41 PM
```

6.4.2 Función time

Permite retornar valores referentes a la fecha y hora actual. Su formato es:

```
time();
```

Veamos el siguiente caso:

- Mostrar la fecha actual y un día posterior a la misma, usando la función **time** para aumentar un día al actual:

```
<?php
    $hoy=date('Y-m-d');
    $mañana = time() + (1 * 24 * 60 * 60);

    echo 'Hoy es:'.$hoy;
    echo '<br>Mañana es: '.date('Y-m-d', $mañana);
?>
```

Resultado:

```
Hoy es:2014-08-01
Mañana es: 2014-08-02
```

Comentarios:

```
$hoy=date('Y-m-d');
```

Declaramos la variable **$hoy**, la cual tiene asignado el día de hoy.

Sólo fines educativos - FreeLibros

```
$mañana = time() + (1 * 24 * 60 * 60);
```

La variable **$mañana** asigna el valor del día siguiente al día actual, sumando 1 día, 24 horas, 60 minutos y 60 segundos.

```
echo 'Hoy es:'.$hoy;
echo '<br>Mañana es: '.date('Y-m-d', $mañana);
```

Los valores son impresos mediante la función **echo** de PHP.

6.4.3 Función checkdate

Permite validar una fecha devolviendo **true** si la fecha es correcta. Su formato es:

```
checkdate($mes, $día, $año);
```

Hay que tener en cuenta que el mes es un valor numérico entre 1 y 12, el día es un valor numérico entre 1 y 30 o 31, dependiendo del mes implementado. El año es un valor numérico entre 1 y 32767.

Veamos el siguiente caso:

- Validar la fecha de registro de un empleado:

```
<?php
    //Asignación de fecha de registro
    $fechaReg='02-26-2015';
    $fecha=explode('-',$fechaReg);

    //Separando día, mes y año
    $mes=$fecha[0];
    $día=$fecha[1];
    $año=$fecha[2];

    //Validando los valores de la fecha
    if (checkdate($mes, $día, $año))
        echo 'La fecha '.$fechaReg. ' es correcta. ';
        else
        echo 'La fecha '.$fechaReg.' NO es correcta..!!';
?>
```

Comentarios:

```
$fechaReg='02-26-2015';
$fecha=explode('-',$fechaReg);
```

Asignamos una fecha de registro en la variable **$fechaReg**, desde aquí desprenderemos el mes, día y año en una variable de tipo arreglo, llamada **$fecha**. La función **explode** permite desprender los elementos y dividir según un criterio; en este caso, por cada guion se estará almacenando un nuevo elemento al arreglo.

Sólo fines educativos - FreeLibros

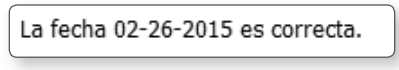
```
$mes=$fecha[0];
$día=$fecha[1];
$año=$fecha[2];
```

Para poder recuperar los valores almacenados en la variable **$fecha**, debemos crear tres variables, las cuales recuperarán la información usando el índice de almacenamiento del arreglo; es decir, el índice 0 indicará el mes, 1 indicará el día y 2 indicará el año.

```
if (checkdate($mes, $día, $año))
```

Finalmente, debemos comprobar que la fecha registrada sea válida usando la función **checkdate**, este recibirá como parámetros de valores individuales de la fecha desprendida desde el arreglo **$fecha**.

Resultado:



6.4.4 Función getdate

Permite generar un arreglo de valores con los elementos que conforman la fecha actual del sistema. Su formato es:

```
getdate(formato);
```

Veamos el siguiente caso:

- A partir de la fecha actual, determinar el día, mes, año, hora, minuto y segundo:

```
<?php
    //Obtener la fecha y hora actual
    $hoy=getdate(time());

    //Datos de la fecha actual
    echo 'Año: '.$hoy['year'];
    echo '<br>Mes: '.$hoy['month'];
    echo '<br>Día: '.$hoy['mday'];

    //Datos de la hora actual
    echo '<br>Hora: '.$hoy['hours'];
    echo '<br>Minutos: '.$hoy['minutes'];
    echo '<br>Segundos: '.$hoy['seconds'];
?>
```

Resultado:

Sólo fines educativos - FreeLibros

6.5 FUNCIONES IMPLEMENTADAS POR EL USUARIO

El código de programación se caracteriza por tener bloques de código que realizan una actividad hasta completar un proceso; normalmente, esta se realiza de forma secuencial, es decir, el compilador interpretará línea por línea de forma **top-down**. Con la incorporación de las estructuras condicionales y repetitivas cambió un poco la perspectiva de desarrollo, ya que no era tan secuencial como al inicio. Es así que cuando termine una aplicación web con PHP notará que ha escrito mucho código y posiblemente código que se repite; es aquí donde entra el término *función*, que permite separar el código en porciones pequeñas, que pueden ser controladas independientemente del código principal.

6.5.1 Definición y usos

Una función se puede definir como un conjunto de instrucciones o sentencias separadas de un código principal para una tarea específica, esto a consecuencia de la cantidad de código que se genera en una aplicación. Finalmente, una función tiene por misión devolver casi siempre un valor resultante.

Siempre se recomienda que antes de comenzar una aplicación, analice los procesos que contiene y asigne un nombre de función, y así poder implementar de la mejor manera posible las aplicaciones, empezaremos por un ejemplo básico:

La tienda de calzados más importante de la ciudad de Lima necesita una aplicación en PHP, la cual permita controlar los pagos que se les realiza a sus vendedores. Se debe tener en cuenta el siguiente criterio:

- Cada vendedor cuenta con un sueldo básico de $250.00.
- Se le asignará un monto adicional al básico, llamado comisión, el cual es obtenido gracias al monto total de ventas que realiza durante un mes, para lo cual debemos considerar la siguiente tabla de categorías:

CATEGORÍA	PORCENTAJE POR COMISIÓN
A	20.00 %
B	16.00 %
C	14.00 %
D	12.00 %

- El sueldo bruto resulta entonces de la adición del sueldo básico más la comisión alcanzada durante el mes.
- En caso resulte el sueldo bruto superior a $2500, se le aplicará un descuento ascendente al 11.50 % del sueldo bruto; caso contrario, no estará afecto a descuento alguno.

La tienda de calzado nos pide que determinemos el monto de la comisión, el monto asignado al sueldo bruto, el monto de descuento y el monto asignado al sueldo neto de un vendedor durante un mes determinado.

Sólo fines educativos - FreeLibros

A continuación, detallaremos las actividades que se tiene que realizar para dar una solución adecuada al caso propuesto, y lo haremos en el siguiente orden:

- Identificar a qué vendedor se le está realizando el pago.
- Determinar el sueldo básico de $250.00.
- Calcular la comisión según el monto vendido y la categoría del vendedor.
- Obtener el monto de ventas que tiene asignado el vendedor en el mes.
- Calcular el monto del sueldo bruto.
- Calcular el monto de descuento según el monto del sueldo bruto.
- Calcular el monto neto asignado al vendedor.

Como notará, hemos agotado todas las actividades que se puede realizar en el caso propuesto; ahora le daremos un orden y le asignaremos un nombre adecuado para la implementación de las funciones:

N.° DE ACTIVIDAD	FUNCIÓN
1	**getVendedor()** Permite capturar el nombre del vendedor que se necesita calcular su sueldo.
2	**getMontoVendido()** Permite obtener el monto vendido por el vendedor mencionado en el paso uno.
3	**determinaSueldoBasico()** Asigna el sueldo básico de $250.00.
4	**calculaComision(monto, categoria)** Calcula la comisión según la categoría, pero como el porcentaje se aplica al monto vendido este último tiene que ser un parámetro para la función.
5	**calculaSueldoBruto(monto,comision)** Permite calcular el sueldo bruto sumando el monto y la comisión obtenida por el vendedor.
6	**calculaDescuento(bruto)** Permite calcular el descuento según el sueldo bruto obtenido.
7	**calculaNeto(bruto,descuento)** Permite calcular el sueldo neto obtenido por el vendedor.

6.5.2 Implementación de una función

Se le llama así cuando se define todas las sentencias o instrucciones necesarias para que la función cumpla con una actividad determinada, es aquí donde colocaremos el código PHP que hemos venido estudiando. El formato para la implementación de una función es:

```
function nombre(Parametros){
    //Sentencias o Instrucciones
    return valor_de_retorno;
}
```

Sólo fines educativos - FreeLibros

Veamos las partes de una función:

- El término **function** es una palabra reservada por el lenguaje PHP para identificar las funciones dentro del compilador.
- El nombre de la función debe cumplir los mismos requisitos del nombre de una variable.
- Los **parámetros** son definidos como variables de entrada que serán usadas estrictamente dentro de la función, y que desde el exterior es enviada como valor; así por ejemplo:
  - function nombre($a): Definición de un solo parámetro para la función.
  - function nombre($a,$b): Definición de dos parámetros para la función.
- Las **sentencias** o **instrucciones** son parte del cuerpo de la función, estas pueden ser cualquier tipo de operación, como asignaciones, expresiones, etc.
- El término **return** permite especificar la salida que tiene la función; es decir, cuál es el resultado emitido por la función, también llamado *factor de devolución*.

Del caso anterior implementaremos las funciones a partir de las actividades ordenadas:

N.° DE ACTIVIDAD	FUNCIÓN
1	**getVendedor(){** return $_POST['txtVendedor']; }
2	**getMontoVendido(){** return $_POST['txtMonto']; **}**
3	**determinaSueldoBasico(){** return 250; **}**

4	**calculaComision($monto, $categoria){**
	if($categoría=='A')
	return 0.2 * $monto;
	elseif($categoría=='B')
	return 0.16 * $monto;
	elseif($categoría=='C')
	return 0.14 * $monto;
	elseif($categoría=='D')
	return 0.12 * $monto;
	else
	return 0;
	}

5	**calculaSueldoBruto($monto,$comision){** return $monto+$comision; **}**

6	**calculaDescuento($bruto){**
	if($bruto>2500)
	return 0.11*$bruto;
	else
	return 0;
	}

7	**calculaNeto($bruto,$descuento){** return $bruto-$descuento; **}**

Sólo fines educativos - FreeLibros

6.5.3 Llamando a una función

La llamada a una función se realiza a consecuencia del uso de las mismas, puesto que toda función debe ser invocada dentro de la aplicación web PHP cuando sea necesario. El formato presenta las siguientes variaciones:

$variable = función();	El resultado de la función sin parámetros es enviado a una variable.
$variable = función(parámetros);	El resultado de una función con parámetros es enviado a una variable.
echo función();	El resultado de una función sin parámetros es impreso directamente en el navegador gracias a la función **echo** del PHP.
echo función(parámetros);	El resultado de una función con parámetros es enviado directamente al navegador por la función **echo**.
$variableR = función()+$variable;	El resultado de la función es usado en una expresión de cualquier tipo, pero su respuesta dependerá de la salida de la misma.

6.5.4 Implementación de una función con parámetros

Seguro se estará preguntando: «¿Cuándo será necesario usar los parámetros?». La respuesta es muy sencilla: los parámetros son valores de entrada a la función; es decir, son valores que son necesarios para completar la función. En realidad, debemos determinar cuáles son los datos que necesita una función para realizar una tarea específica; ellos serán los parámetros. Su formato es:

```
function nombre($parametro1, $parametro2, … $parametroN){
      //Cuerpo de la función
      return valor_retornado;
}
```

Hay que considerar que no hay una cantidad límite de parámetros y que debemos tener cuidado con el orden de envío a los parámetros, en especial los de diferentes tipos de datos.

Veamos algunos casos de implementación de funciones con parámetros:

- Función que determine el promedio de 4 notas de un alumno:

```
function calculaPromedio($nota1, $nota2, $nota3, $nota4){
     return round(($nota1+$nota2+$nota3+$nota4)/4);
}
```

- Función que determine el monto de descuento del subtotal de una venta, en el cual se asume un 10 % de la misma:

```
function calculaDescuento($subtotal){
     $descuento = $subtotal * 0.1;
     return $descuento;
}
```

Sólo fines educativos - FreeLibros

- Función que determine el monto por hora, según la categoría del empleado, sabiendo que A=$25.00, B=$20.00 y C=$15.00; en cualquier otro caso devolver cero como valor de monto:

```
function determinaMontoxHora($categoria){
    if ($categoria=='A')
            return 25.00;
            elseif($categoria=='B')
            return 20.00;
            elseif($categoria=='C')
            return 15.00;
            else
            return 0;
}
```

6.5.5 Implementación de una función con parámetros y con valor por defecto

La definición de un parámetro con valor por defecto indica que cuando se invoque a la función y no se envíe un valor, o simplemente sea vacía, tome el valor por defecto y trabaje en base a dicho valor. Su formato es:

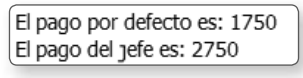
```
function nombre_funcion($parámetro=valor_x_defecto){
      //Cuerpo de la función
      return valor_retornado;
}
```

Veamos el siguiente caso:

- En una empresa los empleados tienen una asignación de pago mensual, dependiendo de la categoría, las cuales son: operario ($750.00), administrativo ($1750.00) y jefe ($2750.00). Implementaremos una función que asigne un monto de pago según su categoría y, además, asignaremos como valor por defecto la categoría «Administrativo».

```
<?php
    //Implementación de la función pago
    function determinaPago($categoria='Administrativo'){
        if ($categoria=='Operario')
            return 750;
        elseif ($categoria=='Administrativo')
            return 1750;
        elseif ($categoria=='Jefe')
            return 2750;
    }

    //Invocando la función
    echo 'El pago por defecto es: '.  determinaPago();
    echo '<br>El pago del Jefe es: '.  determinaPago('Jefe');
?>
```

El resultado de la aplicación es:



Sólo fines educativos - FreeLibros

6.5.6 Implementación de una función sin valor de retorno

Cuando una función no devuelve valor se le llama *función con retorno vacío* o simplemente *sin valor de retorno*; esta forma de implementación se puede usar para imprimir valores directamente. Su formato es:

```
function nombre_funcion(parámetros){
      //Cuerpo de la función
}
```

Veamos el siguiente caso:

- De acuerdo a un monto establecido, calcular el monto de descuento IVA, el cual asciende a un 19 % del monto vendido. Al final deberá imprimir el monto ingresado y el valor del IVA a partir de una función sin valor de retorno.

```
<?php
    //Implementación de la función descuento IVA
    function determinaIVA($monto){
      echo '<br>El monto descuento es: $'
                              .number_format($monto*0.19,'2','.','');
    }

    //Invocando la función sin valor de retorno
    $monto=10000;
    echo 'El monto a pagar es: $'.number_format($monto,'2','.','');
    determinaIVA($monto);
?>
```

Resultado:

El monto a pagar es: $10000.00
El monto descuento es: $1900.00

6.5.7 Implementación de una función con múltiples valores de retorno

Cuando una función emite muchos valores de devolución, estos se tienen que almacenar en un arreglo de valores, el cual debe ser definido dentro de la función. Su formato es:

```
function nombre_funcion(parámetros){
        $variable1=expresion1;
        $variable2=expresion2;
        …
        $variableN=expresionN;
        return array($variable1, $variable2,…,$variableN);
}
```

Veamos el siguiente caso:

- Un alumno cuenta con tres notas, de las cuales se debe determinar el promedio, la nota máxima y la nota mínima; se pide realizar todos los cálculos en una sola función, usando función con múltiples valores de retorno.

Sólo fines educativos - FreeLibros

```php
<?php
    //Implementación de la función
    function estadisticas($n1,$n2,$n3){
        //Calculando el promedio
        $promedio=($n1+$n2+$n3)/3;

        //Determinar el mayor y menor elemento
        $notas=array($n1,$n2,$n3);
        $mayor=max($notas);
        $menor=min($notas);

        //Enviar las respuestas a un arreglo
        return array($promedio, $mayor, $menor);
    }

    //Invocando a la función con múltiples valores
    $nota1=10;
    $nota2=20;
    $nota3=16;

    //Capturando los valores múltiples desde la función
    list($promedio,$mayor,$menor)=estadisticas($nota1,$nota2,$nota3);
    echo 'El promedio es: '.round($promedio);
    echo '<br>La mayor nota es: '.$mayor;
    echo '<br>La menor nota es: '.$menor;
?>
```

Resultado:

El promedio es: 15
La mayor nota es: 20
La menor nota es: 10

6.5.8 Implementación de funciones anónimas (lambda en PHP)

Una función anónima cumple los mismos requerimientos que las funciones comunes; lo que varía es la forma de su implementación, es así que no tiene nombre de función y puede ser usado para implementar función con menos líneas de código.

Una función anónima es una función (o subrutina) definida, y posiblemente invocada, sin estar ligada a un nombre. En el cálculo lambda, todas las funciones son anónimas. El combinador Y puede ser utilizado en estas circunstancias para proveer recursión anónima. Algunos lenguajes de programación también proveen soporte, tanto para funciones con nombre como para las funciones anónimas. Su formato es:

```php
$variable = function(parámetros){
        //Cuerpo de la función
        return valor_de_retorno;
}
```

Veamos el siguiente caso:

- Se tiene un monto $5000.00 de dinero en una cuenta bancaria, simularemos un monto de retiro de $1000.00, el cual debe actualizar a la cuenta por medio de una función anónima (PRIMERA FORMA).

Sólo fines educativos - FreeLibros

```php
<?php
    //Función lambda
    $retiro=function($cuenta,$monto){
        $cuenta-=$monto;
        return $cuenta;
    };

    //Invocando a la función
    $cuenta=5000;
    $monto=1000;
    echo 'El monto de la cuenta es: $'
                .number_format($cuenta,'2','.','');
    echo '<br>El nuevo monto es: $'
                .number_format($retiro($cuenta,$monto),'2','.','');
?>
```

Resultado:

El monto de la cuenta es: $5000.00
El nuevo monto es: $4000.00

- Se tiene un monto $5000.00 de dinero en una cuenta bancaria, simularemos un monto de retiro de $1000.00, el cual debe actualizar a la cuenta por medio de una función anónima (SEGUNDA FORMA).

```php
<?php
    //Función lambda
    $retiro=function($cuenta,$monto){
        $cuenta-=$monto;
        echo '<br>El nuevo monto es: $'
                            .number_format($cuenta,'2','.','');
    };

    //Invocando a la función
    $cuenta=5000;
    $monto=1000;
    echo 'El monto de la cuenta es: $'
                            .number_format($cuenta,'2','.','');
    $retiro($cuenta,$monto);
?>
```

Resultado:

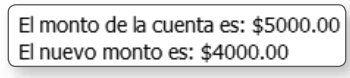
El monto de la cuenta es: $5000.00
El nuevo monto es: $4000.00

- Se tiene un monto $5000.00 de dinero en una cuenta bancaria, simularemos un monto de retiro de $1000.00, el cual debe actualizar a la cuenta por medio de una función anónima (TERCERA FORMA).

Sólo fines educativos - FreeLibros

```php
<?php
 //Función Lambda
 $retiro=create_function('$cuenta,$monto','return $cuenta-=$monto;');

 //Invocando a la función
 $cuenta=5000;
 $monto=1000;
 echo 'El monto de la cuenta es: $'
            .number_format($cuenta,'2','.','');
 echo '<br>El nuevo monto es: $'
            .number_format($retiro($cuenta,$monto),'2','.','');
?>
```

Resultado:



6.6 FUNCIONES INCLUDE Y REQUIRE

Habíamos visto como concepto de función el poder romper el código en porciones pequeñas que trabajan en comunión para un objetivo específico en la aplicación web; pero hasta ahora las funciones implementadas lo han sido dentro de un mismo documento web, aquí radica la ventaja de **include** y **require**, pues dicha función importa código desde el exterior; es decir, permite insertar un código PHP en otro.

Entonces podemos deducir que:

- **Include** o **requiere** invoca código PHP en otro.
- Podemos implementar una aplicación web en varios archivos PHP.
- En una aplicación web, la lógica de negocio puede estar separada del diseño web.
- Tanto **include** como **require** invocan archivos HTML y PHP; otros tipos de archivos deben ser referenciados mediante la etiqueta LINK del HTML.
- Si necesitamos conectarnos a una base de datos podemos incluir una cadena de conexión.

6.6.1 Función include

Es una función que permite importar código PHP dentro de otro, haciendo lo que comúnmente llamamos programación modular. La característica principal de la función **include** es que cuando el archivo especificado no es encontrado muestra un mensaje de advertencia de error más conocido como «Errores warning...!!», su formato es:

```php
include('archivo.php');
include('archivo.html');
```

Veamos el caso de implementación de la función **include**:

Diseñar una aplicación web en PHP que permita controlar la venta de productos en una fuente de soda, en ella se debe ingresar el nombre del cliente y las cantidades de los productos ofrecidos.

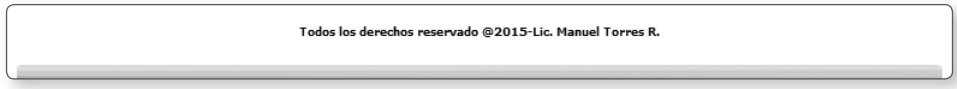
Sólo fines educativos - FreeLibros

La interfaz inicial se muestra a continuación:

Tener en cuenta:

- Debe implementar un archivo llamado **encabezado.php** que permita mostrar la siguiente imagen:

- Debe implementar un archivo llamado **pie.php** que permita mostrar la siguiente imagen:



- Finalmente la aplicación debe mostrar los siguientes resultados:

Sólo fines educativos - FreeLibros

Archivo: **fuente.php**

```
<!DOCTYPE html>
<html>
<head>
   <meta charset="UTF-8">
   <title></title>
   <link href="estilo.css" rel="stylesheet">
</head>
<body>
   <header>
       <?php include('encabezado.php'); ?>
   </header>
   <section>
     <?php
        //Omitiendo errores de advertencia e incluyendo el archivo
        //asignaPrecios.php
        error_reporting(0);
        include('asignaPrecios.php');
     ?>
   <form name="frmHelados" action="helados.php" method="POST">
      <table border="1" width="770"
             cellspacing="1" cellpadding="1">
        <tr>
          <td>Cliente</td>
          <td colspan="3">
              <input type="text" name="txtCliente" size="104"
                     value="<?php echo $cliente; ?>"  /></td>
        </tr>
        <tr>
          <td id="fila">Lista de Productos</td>
          <td id="fila">Cantidad</td>
```

Sólo fines educativos - FreeLibros

```
            <td id="fila">Precio $</td>
            <td id="fila">Subtotal $</td>
          </tr>
          <tr>
            <td>Ensalada de Frutas</td>
            <td><input type="text" name="txtEnsalada"
                       value="<?php echo $cantidadEns; ?>" /></td>
            <td><?php echo '$'.number_format($ensalada,2); ?></td>
            <td><?php echo '$'. number_format($subtotalEns,2)?></td>
          </tr>
          <tr>
            <td>Jugo de Frutas</td>
            <td><input type="text" name="txtJugo"
                       value="<?php echo $cantidadJug; ?>" /></td>
            <td><?php echo '$'.number_format($jugo,2); ?></td>
           <td><?php echo '$'.  number_format($subtotalJug,2)?></td>
          </tr>
          <tr>
            <td>Helado</td>
            <td><input type="text" name="txtHelado"
                       value="<?php echo $cantidadHel; ?>" /></td>
            <td><?php echo '$'.number_format($helado,2); ?></td>
           <td><?php echo '$'.  number_format($subtotalHel,2)?></td>
          </tr>
          <tr>
            <td>Sandwich</td>
            <td><input type="text" name="txtSandwich"
                       value="<?php echo $cantidadSan; ?>" /></td>
            <td><?php echo '$'.number_format($sandwich,2); ?></td>
           <td><?php echo '$'.  number_format($subtotalSan,2)?></td>
          </tr>
          <tr>
            <td></td>
            <td colspan="3"><input name="btnFinalizar" type="submit"
                                   value="Finalizar Venta" /></td>
          </tr>
              <?php
                  if ($_POST['btnFinalizar']){
              ?>
              <tr>
                  <td></td>
                  <td></td>
                  <td>TOTAL A PAGAR</td>
                  <td id="codigo">$
                      <?php echo number_format($total,2); ?></td>
              </tr>
                  <?php } ?>
              </table>
            </form>
          </section>
          <footer>
              <?php include('pie.php'); ?>
          </footer>
      </body>
  </html>
```

Sólo fines educativos - FreeLibros

Archivo: **encabezado.php**

```
<!DOCTYPE html>
<html>
    <head>
        <meta charset="UTF-8">
        <title></title>
    </head>
    <body>
        <img src="bannerHelados.jpg" width="755" height="200"/>
        <h2 id="centrado">FUENTE DE SODA</h2>
    </body>
</html>
```

Archivo: **pie.php**

```
<!DOCTYPE html>
<html>
    <head>
        <meta charset="UTF-8">
        <title></title>
    </head>
    <body>
        <h6 id="centrado">Todos los derechos reservado
                          @2015-Lic. Manuel Torres R.</h6>
        <img src="bannerPie.png" width="770" height="20"/>
    </body>
</html>
```

Archivo: **calculos.php**

```
<?php
    //Asignación de precios
    $ensalada=10;
    $jugo=13;
    $helado=7;
    $sandwich=25;

    //Capturando el nombre del cliente
    $cliente = $_POST['txtCliente'];

    //Capturando las cantidades solicitadas por el usuario
    $cantidadEns = $_POST['txtEnsalada'];
    $cantidadJug = $_POST['txtJugo'];
    $cantidadHel = $_POST['txtHelado'];
    $cantidadSan = $_POST['txtSandwich'];

    //Calculando los subtotales
    $subtotalEns = $ensalada * $cantidadEns;
    $subtotalJug = $jugo * $cantidadJug;
    $subtotalHel = $helado * $cantidadHel;
    $subtotalSan = $sandwich * $cantidadSan;

    //Calculando el total
    $total = $subtotalEns+$subtotalHel+$subtotalSan+$subtotaljug;
?>
```

Sólo fines educativos - FreeLibros

6.6.2 Función require

Tiene la misma funcionalidad que **include** con la diferencia de que no muestra mensaje de advertencias cuando el archivo solicitado no se encuentra, sino más bien detiene la aplicación mostrando un mensaje de «PHP error fatal..!!», ya que por concepto de la palabra **require**, podemos deducir que un código es requerido por otro; por lo tanto, obligatorio, su formato es:

```
require('archivo.php');
require('archivo.html');
```

Veamos el caso de implementación de la función **require**:

Diseñar una aplicación web en PHP que permita controlar el acceso de los usuarios al sistema, de tal forma que se pueda seleccionar un tipo de usuario (administrador, sistemas y operador) con sus respectivas claves de acceso; mostrando así un mensaje de «Usuario logueado correctamente...!!» si el acceso es correcto.

La interfaz inicial se muestra a continuación:

Tener en cuenta:

- Debe implementar un archivo llamado **encabezado.php** que permita mostrar la siguiente imagen:

- Debe implementar un archivo llamado **pie.php** que permita mostrar la siguiente imagen:

Sólo fines educativos - FreeLibros

- Finalmente, la aplicación debe mostrar los siguientes resultados:

Archivo: **principal.php**

```
<!DOCTYPE html>
<html>
<head>
  <meta charset="UTF-8">
  <title></title>
  <link href="estilo.css" rel="stylesheet">
</head>
<body>
  <header>
     <?php require('encabezado.php'); ?>
  </header>
  <section>
     <?php
        error_reporting(0);
        require('captura.php');
        $usuario=  getUsuario();
        $clave=  getPassword();
     ?>
     <form action="principal.php" method="POST">
       <table border="1" width="550"
              cellspacing="1" cellpadding="1">
         <tr>
           <td>Usuario</td>
           <td>
            <select name="selUsuarios">
              <option value="Administrador">Administrador</option>
              <option value="Sistemas">Sistemas</option>
              <option value="Operador">Operador</option>
            </select>
           </td>
         </tr>
         <tr>
           <td>Password</td>
           <td><input type="password" name="txtPassword"
                      maxlength="5" value="" /></td>
         </tr>
         <tr>
           <td colspan="2" id="centrado">
               <input type="submit" value="INGRESAR"
                      name="btnIngresar" />
           </td>
         </tr>
         <tr>
           <td colspan="2" id="centrado">
              <?php
                 if (isset($_POST['btnIngresar'])){
```

Sólo fines educativos - FreeLibros

```
                    require('validar.php');
                    if (valida($usuario, $clave)=='ok')
                      header('location:bienvenida.php');
                      else{
                      echo 'Error de datos..!!';
                      }
                  }
              ?>
            </td>
          </tr>
        </table>
      </form>
 </section>
 <footer>
     <?php require('pie.php'); ?>
 </footer>
</body>
</html>
```

Archivo: **encabezado.php**

```
<!DOCTYPE html>
<html>
    <head>
        <meta charset="UTF-8">
        <title></title>
    </head>
    <body>
      <img src="banner_usuarios.jpg"/>
      <h2 id="centrado">CONTROL DE USUARIOS</h2>
    </body>
</html>
```

Archivo: **pie.php**

```
<!DOCTYPE html>
<html>
    <head>
        <meta charset="UTF-8">
        <title></title>
    </head>
    <body>
      <p id="centrado">
        <a href="principal.php">Principal</a> |
        <a href="javascript:close()">Salir</a>
      </p>
    </body>
</html>
```

Archivo: **captura.php**

```
<?php
    function getUsuario(){
        return $_POST['selUsuarios'];
    }
    function getPassword(){
        return $_POST['txtPassword'];
    }
?>
```

Sólo fines educativos - FreeLibros

Archivo: **validar.php**

```
<?php
 function valida($usuario,$clave){
   $acceso='';
   if ($usuario=='Administrador' && $clave=='1234A') $acceso='ok';
   if ($usuario=='Sistemas' && $clave=='1111A') $acceso='ok';
   if ($usuario=='Operador' && $clave=='2222A') $acceso='ok';

   return $acceso;
 }
?>
```

Archivo: **bienvenida.php**

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="UTF-8">
    <title></title>
    <link href="estilo.css" rel="stylesheet">
  </head>
  <body>
    <header>
       <?php require('encabezado.php'); ?>
    </header>
    <section>
      <table border="1" width="550"
             cellspacing="1" cellpadding="1">
        <tr>
           <td id="centrado">Usuario logeado correctamente.!!</td>
        </tr>
      </table>
    </section>
    <footer>
       <?php require('pie.php'); ?>
    </footer>
   </body>
</html>
```

Archivo: **estilo.css**

```
body{
    font-family: tahoma;
    font-size: 14px;
}
#centrado{
    text-align: center;
}
table {
    margin: auto;
}
img{
    margin: auto;
    display: block;
}
#centrado{
    text-align: center;
```

Sólo fines educativos - FreeLibros

```
}
td {
    border: solid 1px #006699;
    border-top: #006699;
    border-right: #006699;
    border-left: #006699;
    border-bottom: #006699;
}
```

6.7 CASOS DESARROLLADOS

Para todos los casos desarrollados usaremos un solo archivo de estilos, el cual detallamos a continuación:

Archivo: **estilo.css**

```
body{
    font-family: tahoma;
    font-size: 14px;
}
#centrado{
    text-align: center;
}
table {
    margin: auto;
}
img{
    margin: auto;
    display: block;
}
#fila{
    font-size: 13px;
    background: #b9c9fe;
    color: #039;
}
#error{
    color: red;
    font-size: 12px;
}
#codigo{
    color: blue;
    font-size: 36px;
}
```

- **Caso desarrollado 1: Funciones de cadena – Registro de empleados**

  Implemente una aplicación web con PHP que permita generar un código nuevo a un determinado empleado catalogado como nuevo en la empresa, para lo cual deberá registrar sus apellidos, nombres, fecha de nacimiento, estado civil y tipo de sexo.

Sólo fines educativos - FreeLibros

La interfaz gráfica inicial es la siguiente:

Tener en cuenta:

- La aplicación no deberá mostrar mensajes de error.
- Se debe validar los valores ingresados para los apellidos, nombres y fecha de nacimiento, ya que son necesarios para la generación de un código de empleado.
- Al presionar el botón **Autogenerar código** deberá generar el código del empleado usando el siguiente criterio:

14	1	2	38
Año actual	Estado civil 1: Soltero 2: Casado 3: Viudo 4: Divorciado	Sexo 1: Masculino 2: Femenino	Edad Calculada a partir de la fecha de nacimiento

- El estado civil del empleado debe registrarse en un control tipo SELECT con los siguientes elementos: **Soltero**, **Casado**, **Viudo** y **Divorciado**.
- El tipo de sexo del empleado deberá seleccionarse desde un control tipo RADIO.
- Deberá implementar un botón de **Autogenerar código**, el cual permitirá mostrar el nuevo código del empleado de la siguiente manera: 141238.

Sólo fines educativos - FreeLibros

- Finalmente los resultados deben mostrarse de la siguiente forma:

Archivo: **codigo.php**

```
<!DOCTYPE html>
<html>
    <head>
        <meta charset="UTF-8">
        <title>Registro de Empleados</title>
        <link href="estilo.css" rel="stylesheet">
    </head>
    <body>
        <header>
            <h2 id="centrado">Formulario de registro de empleados</h2>
            <img src="bannerEmpleados.jpg" width="750" height="214" />
        </header>
        <section>
        <?php
            error_reporting(0);
            $apellidos=$_POST['txtApellidos'];
            $nombres=$_POST['txtNombres'];
            $fecnac=$_POST['txtFecnac'];
            $estado=$_POST['selEstado'];
            $sexo=$_POST['rdSexo'];

            $mNombres='';
            $mApellidos='';
            $mFecha='';
            $permitidos = '/^[A-Z üÜáéíóúÁÉÍÓÚñÑ]{1,100}$/i';
            if (!preg_match($permitidos,$apellidos)
                                        || !is_string($apellidos))
                $mApellidos ='Registre apellidos..!!';
```

Sólo fines educativos - FreeLibros

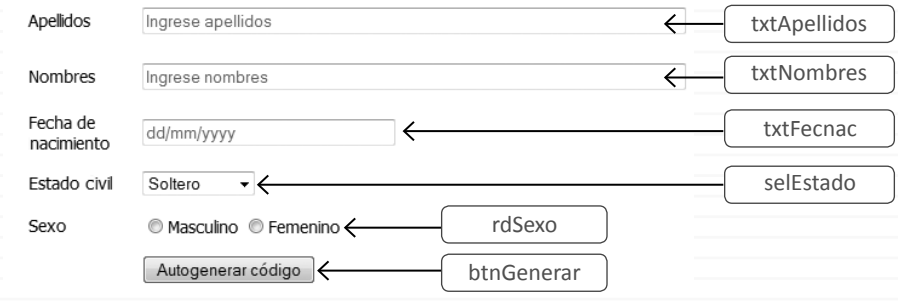
```
        if (!preg_match($permitidos,$nombres)
                                    || !is_string($nombres))
            $mNombres ='Registre nombres..!!';
        if (!preg_match('/^\d{1,2}\/\d{1,2}\/\d{4}$/', $fecnac))
                $mFecha='Fecha no valida';

        if ($estado=='Soltero') $selS='SELECTED'; else $selS="";
        if ($estado=='Casado') $selC='SELECTED'; else $selC="";
        if ($estado=='Viudo') $selV='SELECTED'; else $selV="";
        if ($estado=='Divorciado') $selD='SELECTED'; else $selD="";

        switch($estado){
            case 'Soltero':$cEstado=1;break;
            case 'Casado':$cEstado=2;break;
            case 'Viudo':$cEstado=3;break;
            case 'Divorciado':$cEstado=4;
        }

        if($sexo=='M') $cSexo=1;
        if($sexo=='F') $cSexo=2;

        $aFecha = explode('/', $fecnac);
        $año = $aFecha[2];
        $edad=date('Y')-$año;
    ?>
        <form name="frmRegistro" method="POST" action="codigo.php">
        <table border="0" width="750" cellspacing="0" cellpadding="0">
        <tr>
            <td>Apellidos</td>
            <td><input type="text" name="txtApellidos" size="70"
                       placeholder="Ingrese apellidos"
                       value="<?php echo $apellidos; ?>" /></td>
            <td id="error"><?php echo $mApellidos; ?></td>
            <td>CODIGO GENERADO</td>
        </tr>
        <tr>
            <td>Nombres</td>
            <td><input type="text" name="txtNombres" size="70"
                       placeholder="Ingrese nombres"
                       value="<?php echo $nombres; ?>" /></td>
            <td id="error"><?php echo $mNombres; ?></td>
            <td id="codigo">
                <?php
                    if(isset($_POST["btnGenerar"]))
                        $codigo=substr(date('Y'), 2).$cEstado.
                                                    $cSexo.$edad;
                        else
                        $código='';

                    echo $código
                ?>
            </td>
        </tr>
        <tr>
            <td>Fecha de Nacimiento</td>
            <td><input type="text" name="txtFecnac" size="30"
                       placeholder="dd/mm/yyyy"
                       value="<?php echo $fecnac; ?>" /></td>
            <td id="error"><?php echo $mFecha; ?></td>
            <td></td>
        </tr>
        <tr>
       <td>Estado Civil</td>
       <td>
```

Sólo fines educativos - FreeLibros

```
            <select name="selEstado">
             <option value="Soltero" <?php echo $selS; ?>>Soltero</option>
             <option value="Casado" <?php echo $selC; ?>>Casado</option>
             <option value="Viudo" <?php echo $selV; ?>>Viudo</option>
             <option value="Divorciado" <?php echo $selD; ?>>
                Divorciado </option>
            </select></td>
            <td></td>
            <td></td>
           </tr>
           <tr>
                <td>Sexo</td>
                <td><input type="radio" name="rdSexo" value="M" />
                                                    Masculino</td>
                <td><input type="radio" name="rdSexo" value="F" />
                                                    Femenino</td>
                <td></td>
           </tr>
           <tr>
                <td></td>
                <td><input type="submit" name="btnGenerar"
                                value="Autogenerar codigo" /></td>
                <td></td>
                <td></td>
            </tr>
           </table>
        </form>
        </section>
        <footer>
            <h6 id="centrado">Todos los derechos reservados –
                Lic. Manuel Torres</h6>
        </footer>
    </body>
</html>
```

Comentarios:

```
error_reporting(0);
$apellidos=$_POST['txtApellidos'];
$nombres=$_POST['txtNombres'];
$fecnac=$_POST['txtFecnac'];
$estado=$_POST['selEstado'];
$sexo=$_POST['rdSexo'];
```



Omitimos los errores de advertencia de PHP con la función **error_reporting(0)**, luego capturamos los valores ingresados y seleccionados por el usuario desde los controles del formulario. Debemos tener en cuenta que para obtener los valores seleccionados desde el control radio (para las opciones de sexo) se maneja de manera similar a los demás controles.

Sólo fines educativos - FreeLibros

```
$mNombres='';
$mApellidos='';
$mFecha='';
$permitidos = '/^[A-Z üÜáéíóúÁÉÍÓÚñÑ]{1,100}$/i';

if (!preg_match($permitidos,$apellidos) || !is_string($apellidos))
                $mApellidos ='Registre apellidos..!!';
if (!preg_match($permitidos,$nombres) || !is_string($nombres))
                $mNombres ='Registre nombres..!!';

if (!preg_match('/^\d{1,2}\/\d{1,2}\/\d{4}$/', $fecnac))
                $mFecha='Fecha no valida';
```

Iniciaremos la validación de los valores registrados por el usuario declarando variables para cada elemento validado; luego solo para el caso de los nombres y apellidos se declara la variable **$permitidos**, que contendrá los valores permitidos para el ingreso del nombre y apellido del empleado, esto consta de caracteres en mayúsculas y minúsculas.

Para ambos casos, se preguntará si el valor ingresado es diferente al formato definido como patrón en la variable **$permitidos**; entonces, se emitirá un mensaje de error asignado a la variable **mApellidos** y **mNombres** respectivamente; inclusive hemos agregado la **ñ** y las tildes para apellidos y nombres que lo contengan; la función **preg_match** permite encontrar coincidencias entre los caracteres; y la función **is_string** es para comprobar que se trata de una cadena de texto.

Para la fecha de nacimiento se evalúa un formato directamente en la función **preg_match**, el cual solo permitirá el ingreso de valores con el formato dd/mm/aaaa.

```
if ($estado=='Soltero') $selS='SELECTED'; else $selS="";
if ($estado=='Casado') $selC='SELECTED'; else $selC="";
if ($estado=='Viudo') $selV='SELECTED'; else $selV="";
if ($estado=='Divorciado') $selD='SELECTED'; else $selD="";
```

Verificamos qué tipo de estado civil seleccionó el usuario y se asigna por defecto la opción SELECTED, para que se mantenga dicho valor al momento de generar el código del empleado.

```
switch($estado){
     case 'Soltero':$cEstado=1;break;
     case 'Casado':$cEstado=2;break;
     case 'Viudo':$cEstado=3;break;
     case 'Divorciado':$cEstado=4;
}
```

Ahora evaluamos el estado civil del empleado para asignar un número por cada tipo de estado, y así componer el nuevo código del empleado, según la especificación del caso; 1 será asignado al estado **soltero**, 2 al estado **casado** y así sucesivamente.

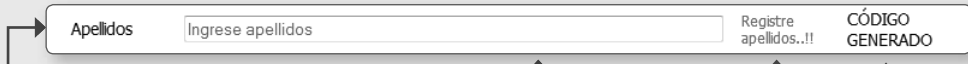
Sólo fines educativos - FreeLibros

```
if($sexo=='M') $cSexo=1;
if($sexo=='F') $cSexo=2;

$aFecha = explode('/', $fecnac);
$año = $aFecha[2];
$edad=date('Y')-$año;
```

De la misma manera que el estado civil, para el tipo de sexo también se le asignará un número: 1 para **masculino** y 2 para **femenino**; hay que tener en cuenta que la variable **$cSexo** formará parte del nuevo código del empleado.

Parte del código autogenerado solicita el año actual, por tanto, debemos obtener la fecha actual en la variable **$aFecha**, luego capturamos el año de dicha fecha en la variable **$año** y, finalmente, calculamos la edad con la fórmula **date('Y')-$año**. Las variables **$año** y **$edad** serán parte del nuevo código del empleado.

Analizaremos el código embebido de HTML5 y PHP para el registro de los apellidos del empleado; no se olvide que también debe ser validado y que se debe mostrar el mensaje de error de color rojo:




```
<tr>
  <td>Apellidos</td>
  <td><input type="text" name="txtApellidos" size="70"
                         placeholder="Ingrese apellidos"
                         value="<?php echo $apellidos; ?>" /></td>
  <td id="error"><?php echo $mApellidos; ?></td>
  <td>CODIGO GENERADO</td>
</tr>
```

La propiedad **placeholder** permite imprimir un mensaje al usuario mediante el control **text.** Si consideramos que esta fila consta de tres columnas y que en la tercera deberá mostrarse un mensaje de error de acuerdo a la validación realizada a los controles del formulario; para este caso, imprimiremos la variable **$mApellidos**, que mostrará el mensaje de error cuando el usuario deje vacío el control o llene con caracteres no permitidos.

Sólo fines educativos - FreeLibros

```
<tr>
    <td>Nombres</td>
    <td><input type="text" name="txtNombres" size="70"
                           placeholder="Ingrese nombres"
                           value="<?php echo $nombres; ?>" /></td>
    <td id="error"><?php echo $mNombres; ?></td>
    <td id="codigo">
          <?php
              if(isset($_POST["btnGenerar"]))
                   $código=substr(date('Y'), 2).$cEstado.$cSexo.$edad;
                   else
                   $código='';

                   echo $codigo
           ?>
     </td>
</tr>
```

En esta fila de la tabla se debe imprimir el código autogenerado del empleado; hemos validado la selección del botón **generar** por parte del usuario, ya que al presionarlo deberá generar el código y registrarlo en la variable **$codigo**. La función **substr** permite obtener una parte del año actual, pues la función **date('Y')** emite 2014, pero en el código solo necesitamos 14; por tanto, substraemos los dos últimos dígitos. Finalmente, concatenamos las variables obtenidas en los pasos anteriores en la variable **$codigo** y se imprimirá en dicha celda.

```
<tr>
   <td>Fecha de Nacimiento</td>
   <td><input type="text" name="txtFecnac" size="30"
                           placeholder="dd/mm/yyyy"
                           value="<?php echo $fecnac; ?>" /></td>
   <td id="error"><?php echo $mFecha; ?></td>
   <td></td>
 </tr>
```

En la tercera columna imprimimos el mensaje de error obtenido a partir de la validación de la fecha de nacimiento del empleado, ello mediante la variable **$mFecha**. Recuerde que en todos los mensajes se invoca a un estilo llamado error que permite mostrar de color rojo el mismo.

```
<tr>
    <td>Estado Civil</td>
    <td>
        <select name="selEstado">
             <option value="Soltero" <?php echo $selS; ?>>Soltero</option>
             <option value="Casado" <?php echo $selC; ?>>Casado</option>
             <option value="Viudo" <?php echo $selV; ?>>Viudo</option>
             <option value="Divorciado" <?php echo $selD; ?>>Divorciado </option>
        </select>
    </td>
    <td></td>
    <td></td>
</tr>
```

La impresión realizada en cada opción permite mantener el estado seleccionado en el control aun después de haber generado el nuevo código del empleado; estas variables fueron asignadas con el valor SELECTED en pasos anteriores.

Sólo fines educativos - FreeLibros

Caso desarrollado 2: Funciones numéricas – Promedio de notas

Implemente una aplicación web con PHP que permita calcular el promedio de notas de un determinado alumno, el cual ingresará su nombre completo y sus cuatro notas, al final se deberá mostrar el nombre del alumno, promedio, nota más alta, nota más baja y la condición del alumno.

La interfaz gráfica propuesta es la siguiente:

Debemos considerar los siguientes aspectos:

- Al iniciar el formulario no debe mostrar los datos resultantes, estos se mostrarán solo cuando el usuario presione el botón **Promediar.**
- El cálculo del promedio, resulta de la tres mejores notas del alumno.
- Use funciones numéricas para determinar la nota más alta y más baja.
- Debemos validar cada control y mostrar un mensaje de error según lo solicitado por el formulario.

A continuación, se muestra la ejecución y el ingreso de valores al formulario:

Sólo fines educativos - FreeLibros

Archivo: **promedio.php**

```
<!DOCTYPE html>
<html>
    <head>
        <meta charset="UTF-8">
        <title></title>
        <link href="estilo.css" rel="stylesheet">
    </head>
    <body>
        <header>
            <h2 id="centrado">Promedio de notas</h2>
            <img src="bannerEstudiantes.jpg" width="660" height="200"/>
            <br>
        </header>
        <section>
        <?php
            error_reporting(0);
            $alumno=$_POST['txtAlumno'];
            $nota1=$_POST['txtNota1'];
            $nota2=$_POST['txtNota2'];
            $nota3=$_POST['txtNota3'];
            $nota4=$_POST['txtNota4'];

            $mAlumno='';
            $permitidos = '/^[A-Z üÜáéíóúÁÉÍÓÚñÑ]{1,100}$/i';
            if (!preg_match($permitidos,$alumno)
                                        || !is_string($alumno))
                $mAlumno='Registre nombres..!!';
            $mNota1='';
            $mNota2='';
```

Sólo fines educativos - FreeLibros

```
        $mNota3='';
        $mNota4='';
        if(empty($nota1) || !is_numeric($nota1))
            $mNota1='Error en Nota 1';
            elseif($nota1<0 || $nota1>20)
            $mNota1='Error en Nota 1';

        if(empty($nota2) || !is_numeric($nota2))
            $mNota2='Error en Nota 2';
            elseif($nota1<0 || $nota2>20)
            $mNota2='Error en Nota 2';

        if(empty($nota3) || !is_numeric($nota3))
            $mNota3='Error en Nota 3';
            elseif($nota3<0 || $nota3>20)
            $mNota3='Error en Nota 3';

        if(empty($nota4) || !is_numeric($nota4))
            $mNota4='Error en Nota 4';
            elseif($nota4<0 || $nota4>20)
            $mNota4='Error en Nota 4';
?>
<form name="frmPromedio" method="post" action="promedio.php">
 <table width="650" border="1" cellspacing="3" cellpadding="3">
      <tr>
        <td width="80">ALUMNO</td>
        <td width="457" id="error">
                <input type="text" name="txtAlumno" size="60"
                       value="<?php echo $alumno; ?>"/>
                <?php echo $mAlumno; ?>
        </td>
      </tr>
      <tr>
        <td>NOTAS</td>
        <td></td>
      </tr>
      <tr>
        <td colspan="2">
        <table width="650" border="1"
               cellspacing="2" cellpadding="2">
          <tr>
            <td width="179">Nota 1</td>
            <td width="179">
                 <input type="text" name="txtNota1" size="5"
                        value="<?php echo $nota1; ?>" />
            </td>
          <td width="179" id="error"><?php echo $mNota1; ?></td>
            <td width="179">Nota 2</td>
            <td width="179">
                 <input type="text" name="txtNota2" size="5"
                        value="<?php echo $nota2; ?>" />
            </td>
          <td width="179" id="error"><?php echo $mNota2; ?></td>
           </tr>
           <tr>
            <td>Nota 3</td>
            <td><input type="text" name="txtNota3" size="5"
                       value="<?php echo $nota3; ?>" />
            </td>
            <td id="error"><?php echo $mNota3; ?></td>
            <td>Nota 4</td>
            <td><input type="text" name="txtNota4" size="5"
                       value="<?php echo $nota4; ?>" />
            </td>
```

Sólo fines educativos - FreeLibros

```
                    <td id="error"><?php echo $mNota4; ?></td>
                  </tr>
                </table>
                </td>
              </tr>
              <tr>
                <td></td>
                <td><input type="submit" value="Promediar" /></td>
              </tr>
              <?php
                $notas=array($nota1,$nota2,$nota3,$nota4);

                $mayor=max($notas);
                $menor=min($notas);

                $promedio=round(($nota1+$nota2+$nota3+$nota4-$menor)/3,0);

                if ($promedio>=13)
                    $condicion='Aprobado';
                else
                    $condicion='Desaprobado';
              ?>
              <tr>
                <td>Alumno</td>
                <td><?php echo $alumno; ?></td>
              </tr>
              <tr>
                <td>Promedio</td>
                <td><?php echo $promedio; ?></td>
              </tr>
              <tr>
                <td>Nota más alta</td>
                <td><?php echo $mayor; ?></td>
              </tr>
              <tr>
                <td>Nota más baja</td>
                <td><?php echo $menor; ?></td>
              </tr>
              <tr>
                <td>Condicion</td>
                <td><?php echo $condicion; ?></td>
              </tr>
            </table>
          </form>
        </section>
        <footer>
            <h6 id="centrado">Todos los derechos reservados -
                Lic. Manuel Torres</h6>
        </footer>
    </body>
</html>
```

Comentarios:

```
error_reporting(0);
$alumno=$_POST['txtAlumno'];
$nota1=$_POST['txtNota1'];
$nota2=$_POST['txtNota2'];
$nota3=$_POST['txtNota3'];
$nota4=$_POST['txtNota4'];
```

Omitimos los errores de advertencia del PHP, luego capturamos todos los valores registrados por el alumno.

Sólo fines educativos - FreeLibros

```
$mAlumno='';
$permitidos = '/^[A-Z üÜáéíóúÁÉÍÓÚñÑ]{1,100}$/i';
if (!preg_match($permitidos,$alumno) || !is_string($alumno))
     $mAlumno='Registre nombres..!!';
```

Validaremos el ingreso del nombre del alumno creando la variable **$permitidos**, que permitirá almacenar los caracteres permitidos en donde se incluya las «ñ» y las tildes. Luego se condiciona verificando el contenido ingresado con el valor asignado a la variable **$permitidos**. Recuerde que la función **preg_match** permite comparar cadenas, y la función **is_string** evalúa si el contenido de la variable es cadena de caracteres.

```
$mNota1='';
$mNota2='';
$mNota3='';
$mNota4='';
if(empty($nota1) || !is_numeric($nota1))
       $mNota1='Error en Nota 1';
       elseif($nota1<0 || $nota1>20)
       $mNota1='Error en Nota 1';

if(empty($nota2) || !is_numeric($nota2))
       $mNota2='Error en Nota 2';
       elseif($nota1<0 || $nota2>20)
       $mNota2='Error en Nota 2';

if(empty($nota3) || !is_numeric($nota3))
       $mNota3='Error en Nota 3';
       elseif($nota3<0 || $nota3>20)
       $mNota3='Error en Nota 3';

if(empty($nota4) || !is_numeric($nota4))
       $mNota4='Error en Nota 4';
       elseif($nota4<0 || $nota4>20)
       $mNota4='Error en Nota 4';
```

En caso de validar las notas se debe considerar (mediante la función **empty**) que no se encuentre vacía la variable; verificar (mediante la función **is_numeric**) que el valor sea numérico y que dicha nota se encuentre entre 0 y 20.

```
<tr>
  <td width="80">ALUMNO</td>
  <td width="457" id="error">
       <input type="text" name="txtAlumno" size="60"
                 value="<?php echo $alumno; ?>"/><?php echo $mAlumno; ?>
  </td>
</tr>
```

En esta fila mantenemos el valor ingresado en el nombre del alumno imprimiendo su nombre en el control **txtAlumno**; además, se debe mostrar (mediante la variable **$mAlumno**) un mensaje de error evaluado en códigos anteriores.

Sólo fines educativos - FreeLibros

```
<tr>
  <td width="179">Nota 1</td>
  <td width="179">
<input type="text" name="txtNota1" size="5"
           value="<?php echo $nota1; ?>" />
  </td>
  <td width="179" id="error"><?php echo $mNota1; ?></td>
  <td width="179">Nota 2</td>
  <td width="179">
<input type="text" name="txtNota2" size="5"
                            value="<?php echo $nota2; ?>" />
   </td>
   <td width="179" id="error"><?php echo $mNota2; ?></td>
</tr>
```

En esta fila imprimimos los valores obtenidos en la **nota1** y **nota2**, haciendo que se mantengan las notas imprimiéndolas dentro del mismo control **text**, además de imprimir el mensaje de error obtenido en códigos anteriores mediante la variable **$mNota1** y **$mNota2**. Tenga en cuenta que su archivo **estilo.css** implemente el estilo **#error**, el cual contiene la definición de texto en color rojo.

```
<tr>
  <td>Nota 3</td>
  <td><input type="text" name="txtNota3" size="5"
                            value="<?php echo $nota3; ?>" />
  </td>
  <td id="error"><?php echo $mNota3; ?></td>
  <td>Nota 4</td>
  <td><input type="text" name="txtNota4" size="5"
                            value="<?php echo $nota4; ?>" />
  </td>
  <td id="error"><?php echo $mNota4; ?></td>
 </tr>
```

En esta fila imprimimos los valores obtenidos en la **nota3** y **nota4** haciendo que se mantengan las notas, imprimiéndolas dentro del mismo control **text**, además de imprimir el mensaje de error obtenido en códigos anteriores mediante la variable **$mNota3** y **$mNota4**. Tenga en cuenta que su archivo **estilo.css** implemente el estilo **#error**, el cual contiene la definición de texto en color rojo.

```
$notas=array($nota1,$nota2,$nota3,$nota4);

$mayor=max($notas);
$menor=min($notas);

$promedio=round(($nota1+$nota2+$nota3+$nota4-$menor)/3,0);

if ($promedio>=13)
         $condicion='Aprobado';
         else
         $condicion='Desaprobado';
```

Sólo fines educativos - FreeLibros

Para poder determinar la mayor y menor nota hemos implementado una variable llamada **$notas**, la cual almacena las **nota1**, **nota2**, **nota3** y **nota4** , como un solo arreglo; a partir de aquí podemos usar la función **max** para determinar cuál de las notas es la mayor, y lo mismo para la menor nota.

Si las variables contienen los siguientes datos: **nota1=20**, **nota2=17**, **nota3=14** y **nota4=18**, con esta asignación logramos que **$notas** se convierta en un arreglo unidimensional, de tal forma que los valores se almacenen de la siguiente manera:

ARREGLO	VALOR
$notas[0]	20
$notas[1]	17
$notas[2]	14
$notas[3]	18

El promedio es calculado a partir de las notas capturadas en pasos anteriores, eliminando la menor nota obtenida a partir de la función **min**; también debemos tener en cuenta que se debe redondear dicho promedio usando la función **round**.

Finalmente evaluamos el promedio obtenido y le asignamos el valor «Aprobado» y «Desaprobado» a la variable **$condicion**, para luego imprimirlo en sus celdas correspondientes.

- **Caso desarrollado 3: Funciones implementadas por el usuario – Venta de productos**

Implemente una aplicación web con PHP que permita determinar el descuento y neto a pagar por la venta de una determinada pieza de computadora, usando funciones implementadas por el usuario.

La interfaz gráfica inicial es la siguiente:

Sólo fines educativos - FreeLibros

Tener en cuenta:

- Al iniciar el formulario no debe mostrar los datos resultantes, estos se mostrarán solo cuando el usuario presione el botón **Procesar**.
- Al seleccionar un producto de la lista deberá mostrar su precio de forma automática.
- Implemente funciones para captura de valores.
- Implemente funciones para la impresión del número de venta, fecha de venta y hora de la venta.
- Implemente funciones que permitan calcular el subtotal, descuento y neto a pagar.
- Los valores seleccionados para la venta deben mantenerse aun después de presionar el botón **Procesar**.
- Debemos validar el ingreso de la cantidad de productos comprados; en caso se ingrese un valor incorrecto, mostrar un mensaje tal como se muestra en la siguiente imagen.
- El monto de descuento se basa en el siguiente criterio:

SUBTOTAL	% DE DESCUENTO
≤ 300	8 %
> 300 y ≤ 500	10 %
<500	20 %

- Finalmente, si todos los datos son correctos, el botón **Procesar** debe generar la siguiente ventana:

Sólo fines educativos - FreeLibros

Archivo: **venta.php**

```
<!DOCTYPE html>
<html>
<head>
   <meta charset="UTF-8">
   <link href="estilo.css" rel="stylesheet">
   <title>Venta de Productos</title>
</head>
<body>
   <header>
     <h2 id="centrado">VENTA DE PRODUCTOS</h2>
     <h3 id="centrado">Piezas de Computadoras</h3>
     <img src="bannerVenta.jpg" width="700" height="212"/>
   </header>
   <section>
     <form name="frmVenta" method="POST">
       <!--Tabla de Ingreso-->
       <table width="700" border="1" cellspacing="0" cellpadding="0">
         <tr>
           <td width="200"></td>
           <td width="200"></td>
           <td width="150">Nº Venta:</td>
           <td width="50"><?php echo generaCodigo(); ?></td>
         </tr>
         <tr>
           <td></td>
           <td></td>
```

Sólo fines educativos - FreeLibros

```
    <td>Fecha:</td>
    <td><?php echo muestraFecha(); ?></td>
  </tr>
  <tr>
    <td></td>
    <td></td>
    <td>Hora: </td>
    <td><?php echo muestraHora(); ?></td>
  </tr>
  <tr>
  <tr>
     <td>Producto</td>
     <?php
         error_reporting(0);
         list($teclado,$mouse,$impresora,$disco,$lectora)=
                                    productoxDefecto(getProducto());
     ?>
     <td colspan="3">
       <select name="selProducto"
                              onchange="this.form.submit()">
         <option value="Teclado"<?php echo $teclado; ?>>
                        Teclado multimedia</option>
         <option value="Mouse"<?php echo $mouse; ?>>
                        Mouse optico</option>
         <option value="Impresora"<?php echo $impresora; ?>>
                        Impresora color</option>
         <option value="Disco Duro"<?php echo $disco; ?>>
                        Disco Duro</option>
         <option value="Lectora DVD"<?php echo $lectora; ?>>
                        Lectora DVD</option>
        </select>
       </td>
       <td></td>
       <td></td>
     </tr>
     <tr>
       <td>Cantidad</td>
       <td><input type="text" name="txtCantidad" size="10"
                  value="<?php echo getCantidad(); ?>" /></td>
       <td>Precio:</td>
       <td id="código"><?php echo '$'.number_format(
                                    asignaPrecio(getProducto()),2); ?>
       </td>
     </tr>
     <tr>
       <td></td>
       <td></td>
       <td><input type="submit" value="PROCESAR" /></td>
       <td></td>
     </tr>
  </table>

<?php if(!empty($_POST['txtCantidad'])) { ?>
<!--Tabla de Impresión de Resultados-->
<table border="1" width="700" cellspacing="0" cellpadding="0">
    <tr>
        <td width="200">NUMERO:</td>
        <td width="200"><?php echo generaCodigo(); ?> </td>
        <td width="200"></td>
        <td width="200"></td>
    </tr>
    <tr>
```

Sólo fines educativos - FreeLibros

```
                <td>FECHA:</td>
                <td><?php echo muestraFecha(); ?></td>
                <td>HORA:</td>
                <td><?php echo muestraHora(); ?></td>
            </tr>
            <tr>
                <td>PRODUCTO</td>
                <td colspan="3"><?php echo getProducto(); ?></td>
                <td></td>
                <td></td>
            </tr>
            <tr>
                <td>CANTIDAD:</td>
                <td><?php echo getCantidad(); ?></td>
                <td>PRECIO:</td>
                <td><?php echo '$'.number_format(
                                        asignaPrecio(getProducto()),2); ?>
                </td>
            </tr>
            <tr>
                <td></td>
                <td></td>
            </tr>
        </table>

        <table border="1" width="700" cellspacing="0" cellpadding="0">
            <tr>
              <td width="200">SUBTOTAL</td>
              <td width="200">
                <?php
                   $subtotal=calculaSubtotal(asignaPrecio(getProducto()),
                                             getCantidad());
                   echo '$'.number_format($subtotal,'2','.','');  ?>
              </td>
              <td width="200"></td>
              <td width="200"></td>
            </tr>
            <tr>
                <td>DESCUENTO</td>
                <td><?php
                        $descuento = calculaDescuento($subtotal);
                        echo '$'.number_format($descuento,'2','.','');  ?>
                </td>
                <td>NETO A PAGAR</td>
                <td id="código"><?php
                        $neto = calculaNeto($subtotal,$descuento);
                        echo '$'.number_format($neto,'2','.','');  ?>
                </td>
            </tr>
        </table>
        <?php }
             else
                echo "<script languaje='javascript'>
                            alert('Ingrese una cantidad correcta..!!')
                      </script>";
        ?>
        </form>
        </section>
        <footer></footer>
    </body>
</html>
```

Sólo fines educativos - FreeLibros

```php
<?php
    function getProducto(){
        return $_POST['selProducto'];
    }
    function getCantidad(){
        return $_POST['txtCantidad'];
    }
    function generaCodigo(){
        $n = 1;
        return str_pad($n,5,'0',STR_PAD_LEFT);
    }
    function muestraFecha(){
        return date('d-m-y');
    }
    function muestraHora(){
        return date('h:i A');
    }
    function asignaPrecio($producto){
        switch ($producto){
           case 'Teclado': return 20; break;
           case 'Mouse': return 30; break;
           case 'Impresora': return 120; break;
           case 'Disco Duro': return 270; break;
           case 'Lectora DVD': return 20; break;
           default: return 0;
        }
    }

    function productoxDefecto($producto){
        if ($producto=='Teclado') $selT='SELECTED'; else $selT="";
        if ($producto=='Mouse') $selM='SELECTED'; else $selM="";
        if ($producto=='Impresora') $selI='SELECTED'; else $selI="";
        if ($producto=='Disco Duro') $selD='SELECTED'; else $selD="";
        if ($producto=='Lectora DVD') $selL='SELECTED'; else $selL="";
        return array($selT,$selM,$selI,$selD,$selL);
    }

    function calculaSubtotal($precio,$cantidad){
        return $precio*$cantidad;
    }

    function calculaDescuento($subtotal){
        if ($subtotal<=300)
            return $subtotal*0.08;
            elseif ($subtotal<=500)
                return $subtotal*0.1;
            else
                return $subtotal*0.2;
    }

    function calculaNeto($subtotal,$descuento){
        return $subtotal-$descuento;
    }
?>
```

Comentarios:

```php
<td><?php echo generaCodigo(); ?></td>
<td><?php echo muestraFecha(); ?></td>
<td><?php echo muestraHora(); ?></td>
```

Sólo fines educativos - FreeLibros

Imprimiremos los valores obtenidos desde las funciones **generaCodigo** (función que genera el código autonumérico del registro), **muestraFecha** (función que muestra la fecha actual) y **muestraHora** (función que muestra la hora actual).

```
error_reporting(0);
list($teclado,$mouse,$impresora,$disco,$lectora)=productoxDefecto(getProducto());
```

Omitimos los errores de advertencia de PHP con la función **error_reporting**. Hemos implementado una función llamada **productoxDefecto**, que permite devolver una matriz de mensajes de acuerdo al producto seleccionado, que luego serán asignados a las variables **$teclado**, **$mouse**, **$impresora**, **$disco** y **$lectora**. La función **list** que recibe dicha matriz tendrá la misión de asignar a cada variable el valor devuelto desde la función en estricto orden de llegada. Considere que para determinar los mensajes, debemos enviar como parámetro el nombre del producto obtenido desde la función **getProducto**.

```
<tr>
      <td>Cantidad</td>
      <td><input type="text" name="txtCantidad" size="10"
                           value="<?php echo getCantidad(); ?>" /></td>
      <td>Precio:</td>
      <td id="código"><?php echo '$'.number_format(asignaPrecio(getProducto()),2); ?>
      </td>
</tr>
```

El script permite imprimir la cantidad ingresada por el usuario en el mismo control, haciendo que no se pierda el valor registrado. En el caso del precio del producto, se mostrará cuando el usuario seleccione un producto desde el cuadro combinado; por tal motivo, debe implementar el siguiente código en el control **selProducto**: <select name="selProducto" onchange="this.form.submit()">.

```
<?php if(!empty($_POST['txtCantidad'])) { ?>
<!--Tabla de Impresion de Resultados-->
<table border="1" width="700" cellspacing="0" cellpadding="0">
```

Verificamos si el usuario ingresó una cantidad adecuada en el control **txtCantidad**, si es correcto se mostrará la tabla de impresión de resultados; hay que tener en cuenta que la llave se debe cerrar al finalizar la tabla.

```
<tr>
      <td width="200">SUBTOTAL</td>
      <td width="200">
      <?php
           $subtotal=calculaSubtotal(asignaPrecio(getProducto()), getCantidad());
           echo '$'.number_format($subtotal,'2','.','');
      ?>
      </td>
      <td width="200"></td>
      <td width="200"></td>
</tr>
```

Sólo fines educativos - FreeLibros

Calculamos el subtotal a pagar por la compra del producto y lo imprimimos en la celda correspondiente, recuerde que la función **number_format** está preparada para redondear a dos decimales el valor numérico.

```
<tr>
          <td>DESCUENTO</td>
          <td><?php
                        $descuento = calculaDescuento($subtotal);
                        echo '$'.number_format($descuento,'2','.','');  ?>
          </td>
          <td>NETO A PAGAR</td>
          <td id="codigo"><?php
                        $neto = calculaNeto($subtotal,$descuento);
                        echo '$'.number_format($neto,'2','.','');  ?>
          </td>
</tr>
```

De la misma forma, calculamos el descuento, el neto y lo imprimimos en sus celdas correspondientes.

```
}
else
  echo "<script languaje='javascript'>
            alert('Ingrese una cantidad correcta..!!')
        </script>";
```

Cuando la cantidad ingresada no es correcta, se debe emitir un mensaje emergente desde el navegador, para lo cual usamos la función **alert** de JavaScript. Para invocar a esta función debemos usar la etiqueta <script language='javascript'>.

```
function getProducto(){
        return $_POST['selProducto'];
}
function getCantidad(){
        return $_POST['txtCantidad'];
}
function generaCodigo(){
        $n = 1;
        return str_pad($n,5,'0',STR_PAD_LEFT);
}
function muestraFecha(){
        return date('d-m-y');
}
function muestraHora(){
        return date('h:i A');
}
function asignaPrecio($producto){
        switch ($producto){
           case 'Teclado': return 20; break;
           case 'Mouse': return 30; break;
           case 'Impresora': return 120; break;
           case 'Disco Duro': return 270; break;
           case 'Lectora DVD': return 20; break;
           default: return 0;
        }
}
```

Sólo fines educativos - FreeLibros

La función **getProducto** es la encargada de capturar el producto seleccionado por el usuario en la compra de productos. La función **getCantidad** es la encargada de obtener la cantidad de productos comprados por el usuario.

La función **generaCodigo** permite autogenerar numéricamente el número de venta a partir de un número entero, para lo cual usamos la función **str_pad** para repetir los ceros a la izquierda y así obtener el siguiente formato 000001. El número 5 representa la cantidad de repeticiones que tendrá el número cero, al final de la función se define STR_PAD_LEFT para que los ceros se impriman al lado izquierdo de la variable **$n**.

La función **muestraFecha** permite retornar la fecha actual capturada desde el sistema, lo mismo ocurre con la función **muestraHora**, será obtenida desde el sistema.

La función **asignaPrecio** permite asignar un precio al producto seleccionado por el usuario, este deberá ser evaluado a partir de un parámetro.

```
function productoxDefecto($producto){
       if ($producto=='Teclado') $selT='SELECTED'; else $selT="";
       if ($producto=='Mouse') $selM='SELECTED'; else $selM="";
       if ($producto=='Impresora') $selI='SELECTED'; else $selI="";
       if ($producto=='Disco Duro') $selD='SELECTED'; else $selD="";
       if ($producto=='Lectora DVD') $selL='SELECTED'; else $selL="";

       return array($selT,$selM,$selI,$selD,$selL);
}
```

La función **productoxDefecto** tiene la misión de devolver un arreglo de mensajes, asignándole solo a uno de ellos el atributo SELECTED; estos serán impresos en cada opción del cuadro combinado **selProducto**; todo esto para poder mantener el producto seleccionado por el usuario en su mismo control.

```
function calculaSubtotal($precio,$cantidad){
      return $precio*$cantidad;
}
function calculaDescuento($subtotal){
      if ($subtotal<=300)
            return $subtotal*0.08;
            elseif ($subtotal<=500)
                return $subtotal*0.1;
            else
                return $subtotal*0.2;
}
function calculaNeto($subtotal,$descuento){
      return $subtotal-$descuento;
}
```

La función **calculaSubtotal** permite determinar el costo subtotal en base a la cantidad y precio del producto. La función **calculaDescuento** determina el monto de descuento en base al monto subtotal, pues según el criterio el subtotal es inferior a 300, o entre 301 y 500, o superior a 500 se debe asignar un porcentaje de descuento.

Finalmente, la función **calculaNeto** permite calcular el monto neto a pagar por la compra de un producto.

Sólo fines educativos - FreeLibros

Caso desarrollado 4: Funciones anónimas – Pensión de estudiantes

Implemente una aplicación web con PHP que permita determinar la pensión mensual que paga un determinado estudiante, este se obtiene de la siguiente manera, con la evaluación de su promedio ponderado:

PROMEDIO	CATEGORÍA
Superior a 17	A
Entre 14 y 17	B
Entre 12 y 14	C
Inferior a 12	D

Para obtener el monto de la pensión mensual se deberá evaluar la categoría obtenida desde el promedio ponderado, según la siguiente tabla:

CATEGORÍA	MONTO MENSUAL
A	$ 550.00
B	$ 650.00
C	$ 750.00
D	$ 800.00

La interfaz gráfica propuesta es la siguiente:

Tener en cuenta:

- Al iniciar el formulario no debe mostrar los datos resultantes, estos se mostrarán solo cuando el usuario presione el botón **Procesar**.
- Implemente funciones para captura de valores.
- Los valores seleccionados para la venta deben mantenerse aun después de presionar el botón **Procesar**.
- Implemente una función lambda que determine la categoría del estudiante según su promedio.
- Implemente una función lambda que calcule la pensión del estudiante según su categoría.
- Implemente una función lambda que determine la fecha actual.
- Implemente una función lambda que asigne un valor constante de 5 cuotas para todos los estudiantes.

Sólo fines educativos - FreeLibros

A continuación, se muestra la ejecución y el ingreso de valores al formulario:

Archivo: **pension.php**

```
<!DOCTYPE html>
<html>
<head>
   <meta charset="UTF-8">
   <link href="estilo.css" rel="stylesheet">
   <title>Pensión de estudiantes</title>
</head>
<body>
  <header>
     <h2 id="centrado">PENSION DE ESTUDIANTES</h2>
     <h3 id="centrado">Simulador de cuotas</h3>
     <img src="bannerUniversidad.JPG" width="700" height="182" />
  </header>

  <section>
     <?php error_reporting(0); ?>
     <form name="frmPension" method="POST">
        <!--Tabla para ingreso de valores-->
        <table border="0" width="700" cellspacing="5" cellpadding="1">
          <tr>
             <td>Alumno</td>
```

Sólo fines educativos - FreeLibros

```
            <td><input type="text" name="txtAlumno" size="70"
                       value="<?php echo getAlumno(); ?>" /></td>
            <td></td>
         </tr>
         <tr>
            <td>Promedio ponderado</td>
            <td><input type="text" name="txtPromedio"
                     value="<?php echo getPromedio(); ?>" /></td>
            <td><input type="submit" value="PROCESAR"
                       name="btnProcesar"/></td>
         </tr>
      </table>
      <?php
         $categoria = function($promedio){
         if ($promedio>17)
             return "A";
             elseif ($promedio>=14)
             return "B";
             elseif ($promedio>=12)
             return "C";
             else
             return "D";
         };
       ?>

       <?php if(isset($_POST["btnProcesar"])){?>
       <!--Tabla para el resumen de categoria-->
       <table border="1" width="700" cellspacing="1" cellpadding="1">
          <tr>
             <td width="200">Alumno</td>
             <td width="200"><?php echo getAlumno(); ?></td>
             <td width="200"></td>
          </tr>
          <tr>
             <td>Promedio</td>
             <td><?php echo getPromedio(); ?></td>
             <td></td>
          </tr>
          <tr>
             <td>Categoria</td>
             <td><?php echo $categoria(getPromedio()); ?></td>
             <td></td>
          </tr>
       </table>
       <?php
          $pension=  function($categoria){
          if ($categoria=="A")
              return 550;
              elseif ($categoria=="B")
              return 650;
              elseif ($categoria=="C")
              return 750;
              elseif ($categoria=="D")
              return 800;
              else
              return '';
          };

          $fecha = function(){ return date('d/m/Y'); };

          $cuotas = function() { return 5; };
```

Sólo fines educativos - FreeLibros

```
        ?>

        <!--Tabla resumen de pensión-->
        <table border="1" width="700" cellspacing="1" cellpadding="1">
        <tr>
          <td width="200">RESUMEN DE CUOTAS</td>
          <td width="200"></td>
          <td width="200"></td>
        </tr>
        <tr>
          <td>Monto Pensión</td>
          <td><?php echo '$'.number_format($pension($categoria(
                                    getPromedio())),'2','.',''); ?>
          </td>
          <td></td>
        </tr>
        <tr>
          <td>Fecha Actual</td>
          <td><?php echo $fecha(); ?></td>
          <td></td>
        </tr>
        <tr>
          <td>Número de Cuotas</td>
          <td><?php echo $cuotas(); ?></td>
          <td></td>
         </tr>
        </table>

        <!--Tabla para las posibles fechas y cuotas-->
        <table border="1" width="700" cellspacing="1" cellpadding="1">
          <tr>
            <td colspan="2">RESUMEN DE FECHAS Y CUOTAS</td>
          </tr>
          <tr>
            <td>Fechas</td>
            <td>Monto por cuota</td>
          </tr>
          <?php
             for($i=1;$i<=$cuotas();$i++){
                $montoTotal += $pension($categoria(getPromedio()));
          ?>
          <tr>
             <td>
               <?php
                   $f=$fecha();
                   echo date('d/m/Y', strtotime("$f +$i month"));
               ?>
             </td>
             <td><?php echo '$'.number_format($pension($categoria(
                                 getPromedio())),'2','.',''); ?>
             </td>
          </tr>
          <?php } ?>
          <tr>
             <td>Total por semestre</td>
             <td><?php echo '$'.number_format($montoTotal,'2','.','');?>
             </td>
          </tr>
        </table>
        <?php } ?>
      </form>
```

Sólo fines educativos - FreeLibros

```
    </section>
    <footer>
        <h6 id="centrado">Todos los derechos reservados –
            Lic. Manuel Torres</h6>
    </footer>
    <?php
        function getAlumno(){ return $_POST['txtAlumno']; }
        function getPromedio(){ return $_POST['txtPromedio']; }
      ?>
  </body>
</html>
```

Comentarios:

```
$categoria = function($promedio){
if ($promedio>17)
       return "A";
       elseif ($promedio>=14)
       return "B";
       elseif ($promedio>=12)
       return "C";
       else
       return "D";
};
```

Se implementa la función anónima (lambda), que determina la categoría según el promedio obtenido por el alumno, esta función asignará directamente la categoría a la variable **$categoria**.

```
$pension=  function($categoria){
             if ($categoria=="A")
                 return 550;
                 elseif ($categoria=="B")
                 return 650;
                 elseif ($categoria=="C")
                 return 750;
                 elseif ($categoria=="D")
                 return 800;
                 else
                 return '';
                 };

$fecha = function(){ return date('d/m/Y'); };

$cuotas = function() { return 5; };
```

Usamos la función anónima (lambda) para determinar el monto de la pensión en base a la categoría obtenida en el proceso anterior; este resultado será asignado directamente a la variable **$pensión**.

Luego obtenemos la fecha actual por medio de una función anónima (lambda), y le asignamos a la variable **$fecha**. De la misma forma al número de cuotas constante de la aplicación llamada **$cuotas**.

Sólo fines educativos - FreeLibros

```
<?php
      for($i=1;$i<=$cuotas();$i++){
            $montoTotal += $pensión($categoria(getPromedio()));
?>
<tr>
      <td>
       <?php
          $f=$fecha();
          echo date('d/m/Y', strtotime("$f +$i month"));
       ?>
       </td>
       <td><?php echo '$'.number_format($pension($categoria(
                                  getPromedio())),'2','.',''); ?>
       </td>
</tr>
<?php } ?>
```

De acuerdo al número de cuotas se procederá a imprimir los montos en la celda correspondiente, y a la vez se irá acumulando los montos en la variable **$montoTotal**. Aquí es donde imprimiremos los valores enviados desde las funciones anónimas, como la fecha actual y la pensión.

```
function getAlumno(){ return $_POST['txtAlumno']; }
function getPromedio(){ return $_POST['txtPromedio']; }
```

La función **getAlumno** permite obtener el nombre del alumno registrado en el control **txtAlumno**, la función **getPromedio** permite obtener el promedio registrado en el control **txtPromedio**.

Sólo fines educativos - FreeLibros

CAP.

7

Arreglos

Sólo fines educativos - FreeLibros

Sólo fines educativos - FreeLibros

7.1 INTRODUCCIÓN

Hasta el momento los valores almacenados en las variables han tenido un solo valor; es decir, se pudo registrar un solo nombre de cliente o la descripción de un producto. Sabemos que cuando un cliente cuando realiza una compra no lo hace para adquirir un solo producto sino para varios, inclusive de número indeterminado, es aquí donde entra en escena el término **array** o también conocido como **arreglo**, pues este permitirá almacenar mucha información en una sola variable.

Ahora usted se dará cuenta por qué todos los casos desarrollados fueron elaborados en singular. Dejando de lado dicho término, debemos concentrarnos exclusivamente en los elementos que puede contener un arreglo, pues a comparación de otros lenguajes de programación, PHP cuenta con dos tipos de elementos; ahí radica la diferencia, ya que estos usan los arreglos manejados por índices conocido como **arreglo indexado** o por índice, mientras que PHP agrega un tipo de arreglo mucho más interesante llamado **arreglo asociativo**, el cual explicaremos detalladamente en este capítulo.

No piense que el uso de los arreglos harán un sistema completo para una empresa o algo por el estilo; los arreglos manejan información no relacionada, es decir, sin un orden lógico. Eso no es adecuado; si fuera el caso, entonces debemos usar base de datos como MySQL o PostgreSQL.

Nombraremos algunas características de los arreglos, luego usted reflexionará en qué casos podrá usar los arreglos en una aplicación web:

- Los arreglos pueden almacenar grandes porciones de información llamado valores.
- Los arreglos en PHP cuentan con métodos y propiedades que controlan los valores introducidos de la mejor manera.
- La manipulación de los valores se puede realizar directamente o usando estructuras repetitivas.
- Los arreglos controlan su tamaño de forma dinámica; es decir, podremos agregar valores sin preocuparnos del tamaño declarado para el arreglo, pues al ser dinámico aumenta de capacidad conforme le dan valores.

Los arreglos suelen tener diferentes dimensiones de trabajo, es así que se dividen en unidimensional, bidimensional y multidimensional.

Arreglo unidimensional:

10	20	30	40
0	1	2	3

Arreglo bidimensional:

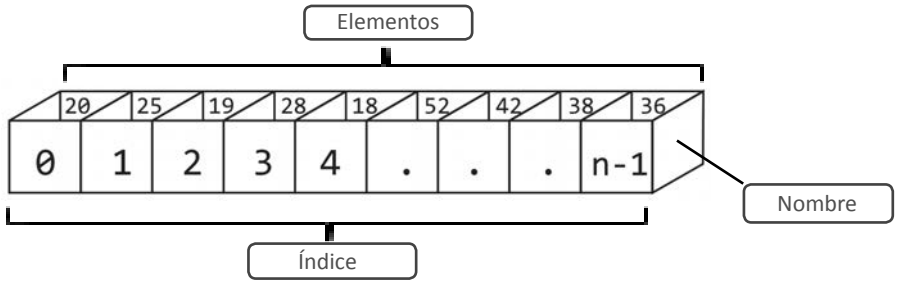
(0,0) 10	(0,1) 20	(0,2) 30
(1,0) 40	(1,1) 50	(1,2) 60
(2,0) 70	(2,1) 80	(2,2) 90

Sólo fines educativos - FreeLibros

7.1.1 Tipos de arreglos

A. Arreglo indexado o por índice

Tiene como característica principal almacenar sus elementos en un orden consecutivo iniciado en cero.



Donde:

- **Elementos:** Son los valores que contiene el arreglo; en este caso podría tratarse de las edades de ciertos empleados en una empresa o del número de horas trabajadas, pero lo que debemos tener en cuenta es que los valores deben ser del mismo tipo.
- **Índice**: Es la posición asignada a cada elemento del arreglo, este siempre empezará en cero y terminará en el número total de elementos menos uno.
- **Nombre**: Es el nombre del arreglo, este debe tratarse como los nombres asignados a las variables.

Veamos el formato para crear un arreglo indexado usando la función **array**:

```
$nombre_arreglo = array(valor1, valor2, valor3, valor4…);
$nombre_arreglo=array(posición=>valor1,valor2,valor3…);
```

La función **array** permite registrar los elementos dentro de un arreglo, de tal forma que el orden de los elementos genera el número de índices del arreglo. Veamos algunos ejemplos de declaración de arreglos:

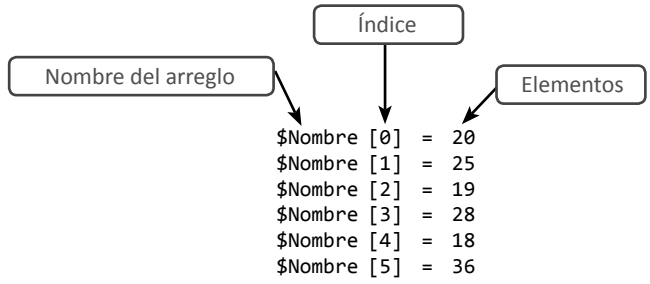
```
$edad=array(20,25,19,28,18,36);
```

De otra manera:

```
$edad=array(0=>20,25,19,28,18,36);
```

Sólo fines educativos - FreeLibros

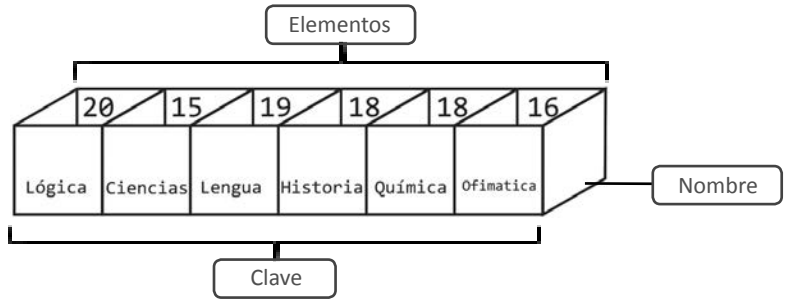
Entonces podemos concluir que la posición de cada elemento del arreglo tomará el siguiente aspecto:



Desde aquí podríamos decir que para mostrar todos los elementos usaremos una estructura repetitiva como **for** o **while**, estas podrán ubicarnos en los índices del arreglo.

B. Arreglo asociativo

Se caracteriza por asignar a cada elemento del arreglo una clave de acceso, algo así como una llave; esta puede tratarse como una cadena identificativa del valor asociado. Parece ser algo complicado de entender, pero los arreglos asociados son los más usados en PHP, por tanto, lo aprenderemos de una u otra forma.



Donde:

- **Elementos**: Son los valores que contiene el arreglo asociado.
- **Clave**: Es la llave para acceder a los valores del arreglo asociado.
- **Nombre**: Es el nombre del arreglo, este debe tratarse como los nombres asignados a las variables.

Veamos el formato para crear un arreglo asociado usando la función **array**:

```
$nombre_arreglo=array(clave1=>valor1, clave2=>valor2, …);
```

La función **array** permite registrar los elementos dentro de un arreglo asociado, de tal forma que la clave es la única forma de poder acceder a los elementos. Veamos la siguiente declaración:

```
$puntaje=array('Fernanda Torres'=>100, 'Ángela Lázaro'=>125,
               'Luz Menor'=>90, 'Lucero Mendoza'=>45);
```

Sólo fines educativos - FreeLibros

Y para probar que el arreglo **puntaje** fue aplicado correctamente, ejecutaremos la siguiente instrucción:

```
print_r($puntaje);
```

Dando como resultado la siguiente expresión:

```
Array ( [Fernanda Torres] => 100 [Ángela Lázaro] => 125
              [Luz Menor] => 90 [Lucero Mendoza] => 45 )
```

Entonces, podemos llegar a concluir que los elementos del arreglo toman el siguiente aspecto:

```
$puntaje['Fernanda Torres'] = 100
$puntaje['Ángela Lázaro'] = 125
$puntaje['Luz Menor'] = 90
$puntaje['Lucero Mendoza'] = 45
```

Desde aquí podríamos decir que para mostrar todos los elementos, usaremos una estructura repetitiva como **foreach**, este permitirá acceder a cada elemento del arreglo mediante una clave. Finalmente, podríamos obtener el puntaje obtenido por la candidata **Luz Menor** para esto debemos invocarlo de la siguiente manera mediante la clave **echo $puntaje['Luz Lázaro']**.

7.2 Estructura repetitiva foreach

Foreach nos permite recorrer únicamente y exclusivamente por los elementos de un arreglo, ya sea asociado o indexado, también puede usarse para recorrer objetos de una clase. Su formato es:

```
foreach ($nombre_array as $valor){
    //Instrucciones repetidas;
}
```

Donde:

- **foreach**: Es la palabra reservada por PHP para la estructura repetitiva.
- **$nombre_array**: Es el nombre del arreglo que contiene la colección de elementos.
- **$valor**: Es la variable que obtiene los valores desde la colección de elementos del arreglo.

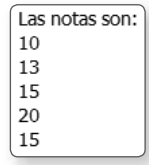
Veamos los siguientes casos:

- Crear un arreglo de notas con los siguientes elementos 10, 13,15, 20 y 15 e imprimirlos:

```
<?php
    //Llenando el arreglo de notas
    $notas=array(10,13,15,20,15);

    //Imprimiendo los elementos
    echo 'Las notas son.<br>';
    foreach ($notas as $n) {
        print $n.'<br>';
    }
?>
```

Sólo fines educativos - FreeLibros

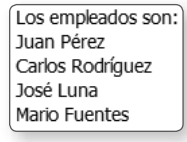
El resultado será:



- Crear un arreglo de empleados con los siguientes elementos **Juan Pérez**, **Carlos Rodríguez**, **José Luna** y **Mario Fuentes**, e imprimirlos:

```
<?php
    //Llenando el arreglo de empleados
    $empleados=array('Juan Pérez','Carlos Rodríguez',
                     'José Luna','Mario Fuentes');

    //Imprimiendo los elementos
    echo 'Los empleados son: <br>';
    foreach ($empleados as $e) {
        print $e.'<br>';
    }
?>
```

El resultado será:



Un segundo formato de la estructura **foreach** usa un puntero que asocia la clave al valor:

```
foreach ($nombre_array as $clave => $valor){
    //Instrucciones repetidas;
}
```

Donde:

- **foreach:** Es la palabra reservada por PHP para la estructura repetitiva.
- **$nombre_array:** Es el nombre del arreglo que contiene la colección de elementos.
- **$clave:** Es el nombre o llave que permite obtener el valor desde el arreglo asociado.
- **$valor:** Es la variable que obtiene los valores desde la colección de elementos del arreglo asociado.

Veamos los siguientes casos:

- Crear un arreglo en el cual se pueda almacenar el número total de ventas realizadas, las cuales contaremos de acuerdo a la siguiente tabla de valores:

EMPLEADO	NÚMERO DE VENTAS
Juan Pérez	100
María López	200
José Ramírez	140
Carlos Diez	110

Sólo fines educativos - FreeLibros

```php
<?php
    //Llenando el arreglo asociativo
    $ventas = array('Juan Pérez' => 100,'Maria López' => 200,
                    'José Ramírez' => 140,'Carlos Diez' => 110);

    //Imprimiendo los elementos
    foreach($ventas as $empleado => $total) {
        echo 'El empleado '.$empleado.'
              tiene un total de '.$total.' ventas realizadas';
        echo '<br>';
    }
?>
```

El resultado es:

El empleado Juan Pérez tiene un total de 100 ventas realizadas
El empleado María López tiene un total de 200 ventas realizadas
El empleado José Ramírez tiene un total de 140 ventas realizadas
El empleado Carlos Diez tiene un total de 110 ventas realizadas

Donde la variable **$empleado** almacenará todos los nombres de los empleados encontrados en el arreglo asociativo. Veamos una segunda forma de impresión del arreglo asociativo, usando la estructura **for**:

```php
<?php
    //Llenando el arreglo asociativo
    $ventas = array('Juan Pérez' => 100,'Maria López' => 200,
                    'José Ramírez' => 140,'Carlos Diez' => 110);

    //Imprimiendo los elementos
    for($i=0;$i<count($ventas);$i++){
        $elementos = each($ventas);
        echo $elementos[0];
        echo $elementos[1].'<br>';
    }
?>
```

La función **each** envía de forma ordenada la pareja clave, valor almacenado en el arreglo; para poder imprimir todas las parejas debemos incluirlas en una estructura repetitiva como **for**. Un detalle particular de las parejas clave-valor es el índice 0 y 1; este índice indica que 0 es clave mientras que 1 es el valor, también pudo ser implementada de la siguiente manera:

```php
<?php
    //Llenando el arreglo asociativo
    $ventas = array('Juan Pérez' => 100,'Maria López' => 200,
                    'José Ramírez' => 140,'Carlos Diez' => 110);

    //Imprimiendo los elementos
    for($i=0;$i<count($ventas);$i++){
        $elementos = each($ventas);
        echo $elementos['key'];
        echo $elementos['value'].'<br>';
    }
?>
```

Sólo fines educativos - FreeLibros

Ahora realizaremos la misma impresión usando la función **list** y la estructura **while**:

```
<?php
    //Llenando el arreglo asociativo
    $ventas = array('Juan Pérez' => 100,'Maria López' => 200,
                    'José Ramírez' => 140,'Carlos Diez' => 110);

    //Imprimiendo los elementos
    while(list($empleado,$total) = each($ventas)){
        echo $empleado;
        echo $total.'<br>';
    }
?>
```

7.3 ADMINISTRAR ELEMENTOS DE UN ARREGLO

La administración de los elementos se refiere al manejo interno del arreglo, como insertar elementos directamente o mediante una función, recorrer los elementos cuando el arreglo es indexado y asociativo.

7.3.1 Insertar elementos

Para insertar un elemento debe considerar que dicho elemento genera una nueva posición del arreglo; si en caso especifica una posición que ya contiene un elemento, entonces solo modificará el contenido. La inserción de elementos dependerá del tipo de arreglo declarado por el usuario. Veamos el formato para un **arreglo indexado**:

```
$nombre_arreglo[ ] = valor o expresión;
$nombre_arreglo[posición] = valor o expresión;
```

Si necesitamos almacenar 5 edades de personas en un arreglo llamado **edad** , entonces tendríamos la siguiente forma de inserción de elementos:

```
$edad[]=15;
$edad[]=24;
$edad[]=32;
$edad[]=25;
$edad[]=50;
```

Cuando no se especifica el índice, automáticamente la posición inicial del arreglo es cero, por tanto, si algún valor es enviado por primera vez, entonces se direccionará a dicha posición. Otra forma de realizarlo sería especificando la posición de cada elemento, como se muestra en el siguiente código:

```
$edad[0]=15;
$edad[1]=24;
$edad[2]=32;
$edad[3]=25;
$edad[4]=50;
```

Sólo fines educativos - FreeLibros

Se podría decir que al indicar la posición del elemento aseguraríamos que el valor se almacena en una determinada posición del arreglo; finalmente, podríamos combinar ambas opciones de la siguiente manera:

```
$edad[]=15;
$edad[]=24;
$edad[2]=32;
$edad[3]=25;
$edad[]=50;
```

Veamos, el valor 15 se almacena en la posición 0; 24 en la posición 1; 32 en la posición especificada 2; 25 en la posición especificada 3, y finalmente el valor 50 en la posición 4.

Si necesitamos imprimir los valores almacenados en el arreglo podríamos usar el siguiente script:

```
echo 'Los elementos del arreglo son: ';
print_r($edad);
```

Dando como resultado la siguiente expresión:

```
Array ( [0] => 15 [1] => 24 [2] => 32 [3] => 25 [4] => 50 )
```

Para la inserción de elementos en un **arreglo asociativo** debemos seguir el siguiente formato:

```
$nombre_arreglo ['clave'] = valor o expresión;
```

Si necesitamos almacenar cinco montos de ventas tal como se muestra en la siguiente tabla:

Empleado	Monto de Venta $
Juan Pérez	10000.00
María López	5500.00
José Ramírez	12000.00
Carlos Diez	6500.00

El script sería de la siguiente manera:

```
$monto['Juan Pérez']=10000;
$monto['María López']=5500;
$monto['José Ramírez']=12000;
$monto['Carlos Diez']=6500;
```

Ahora, si necesitamos imprimir los valores de dicho arreglo, podemos usar el siguiente script:

```
echo 'Las ventas son: ';
echo print_r($monto);
```

Dando como resultado la siguiente expresión:

```
Array ( [Juan Pérez] => 10000 [María López] => 5500 [José Ramírez] => 12000 [Carlos
Diez] => 6500 )
```

Sólo fines educativos - FreeLibros

7.3.2 Insertar elementos numéricos mediante una función

Un arreglo puede contener valores numéricos de forma consecutiva usando la función **range**, esta provee de dos elementos de inserción que indican el rango de inicio y fin de los valores a almacenar en el arreglo. Hay que tener en cuenta que las posiciones son determinadas a partir de cero, su formato es:

```
$arreglo  = range(valor_inicial,valor_final);
```

Donde:

- **Valor inicial**: Representa el valor inicial que se almacenará en la posición cero del arreglo.
- **Valor final**: Representa el valor final que se almacenará en la posición que tiene la siguiente fórmula **Valor_final-Valor_inicial**.

Veamos algunos casos:

- Script que permita registrar los números entre 10 y 20 en un arreglo numérico, el cual imprima el índice y el elemento almacenado usando la función **range**:

```
<?php
    //Llenando el arreglo con la función range
    $arreglo = range(10,20);

    //Imprimiendo los elementos
    echo '<pre>';
    echo "Indice \t    Elemento<br>";
    for($i=0;$i<count($arreglo);$i++){
        $elementos = each($arreglo);
        echo $elementos[0]."\t\t";
        echo $elementos[1].'<br>';
    }
    echo '</pre>';
?>
```

El resultado es:

```
Índice      Elemento
0               10
1               11
2               12
3               13
4               14
5               15
6               16
7               17
8               18
9               19
10              20
```

Sólo fines educativos - FreeLibros

7.3.3 Recorrer los elementos por índice

Siempre que necesitemos recorrer por los valores de un arreglo indexado, debemos usar una estructura repetitiva, veamos el caso en que debemos imprimir el siguiente arreglo:

```
$edad[]=15;
$edad[]=24;
$edad[]=32;
$edad[]=25;
$edad[]=50;
```

O también expresado como: `$edad = array(15,24,32,25,50);`

Para imprimir dichos valores, tal como se muestra en el siguiente formato:

Posición	Edad
0	15
1	24
2	32
3	25
4	50

El script para solución el caso sería:

```
<!DOCTYPE html>
<html>
    <head>
        <meta charset="UTF-8">
        <title>Recorrer un arreglo por índice</title>
    </head>
    <body>
        <?php
            $edad[]=15;
            $edad[]=24;
            $edad[]=32;
            $edad[]=25;
            $edad[]=50;
        ?>
        <table border="1" width="400" cellspacing="0" cellpadding="0">
            <tr>
                <td>Posición</td>
                <td>Edad</td>
            </tr>
            <tr>
                <td>0</td>
                <td><?php echo $edad[0]; ?></td>
            </tr>
            <tr>
                <td>1</td>
                <td><?php echo $edad[1]; ?></td>
            </tr>
            <tr>
                <td>2</td>
                <td><?php echo $edad[2]; ?></td>
            </tr>
            <tr>
                <td>3</td>
                <td><?php echo $edad[3]; ?></td>
            </tr>
```

Sólo fines educativos - FreeLibros

```
            <tr>
                <td>4</td>
                <td><?php echo $edad[4]; ?></td>
            </tr>
        </table>
    </body>
</html>
```

Como notará, el código será más extenso si registramos más edades en el arreglo; es aquí donde se usa la estructura repetitiva que permitirá administrar menos líneas de código, de tal forma que el script será de la siguiente manera:

```
<!DOCTYPE html>
<html>
    <head>
        <meta charset="UTF-8">
        <title>Recorrer un arreglo por índice</title>
    </head>
    <body>
        <?php
            $edad[]=15;
            $edad[]=24;
            $edad[]=32;
            $edad[]=25;
            $edad[]=50;
        ?>
        <table border="1" width="400" cellspacing="0" cellpadding="0">
            <tr>
                <td>Posición</td>
                <td>Edad</td>
            </tr>
            <?php
                for($i=0;$i<count($edad);$i++){
            ?>
            <tr>
                <td><?php echo $i; ?></td>
                <td><?php echo $edad[$i]; ?></td>
            </tr>
            <?php } ?>
        </table>
    </body>
</html>
```

Observemos que la cantidad de registro de edades puede aumentar o disminuir dentro del arreglo, mientras que el código de impresión de elementos no variará, ya que está controlado por una estructura repetitiva.

La función **count** especificada en la estructura **for**; permite determinar el total de elementos registrados en el arreglo, pero recuerde que el arreglo tiene un punto de inicio de índice en cero; por tanto, el valor devuelto por la función **count** debe ser siempre un valor menos, de tal manera que podríamos tener dos versiones del mismo **for**:

```
for($i=0;$i<=count($edad)-1;$i++)
for($i=0;$i<count($edad);$i++)
```

Sólo fines educativos - FreeLibros

O también podríamos usar la estructura **while** de la siguiente manera:

```
<!DOCTYPE html>
<html>
    <head>
        <meta charset="UTF-8">
        <title>Recorrer un arreglo por índice</title>
    </head>
    <body>
        <?php
            $edad[]=15;
            $edad[]=24;
            $edad[]=32;
            $edad[]=25;
            $edad[]=50;
        ?>
        <table border="1" width="400" cellspacing="0" cellpadding="0">
            <tr>
                <td>Posición</td>
                <td>Edad</td>
            </tr>
            <?php
                $n=count($edad);
                $i=0;
                while($i<$n){
            ?>
            <tr>
                <td><?php echo $i; ?></td>
                <td><?php echo $edad[$i]; ?></td>
            </tr>
            <?php
                $i++;
                }
            ?>
        </table>
    </body>
</html>
```

Donde **$n=count($edad)** determina el total de elementos del arreglo, **$i** tiene por misión iniciar el ciclo de repeticiones con **while**; no olvide asignar un valor más uno a la variable **$i** para dar ciclos de repeticiones **while**.

En todos los casos el resultado es:

Posición	Edad
0	15
1	24
2	32
3	25
4	50

7.3.4 Recorrer por elementos asociativos

Cuando los elementos son asociados, la forma de recorrido varía a comparación del arreglo indexado. Hay que tener en cuenta que debemos conocer el formato de la estructura **foreach**, ya que este tipo de arreglo no depende del índice sino de la clave. Si tenemos los puntajes obtenidos en un certamen de belleza:

```
$puntaje=array('Fernanda Torres'=>100, 'Ángela Lázaro'=>125,
                'Luz Menor'=>90, 'Lucero Mendoza'=>45);
```

Sólo fines educativos - FreeLibros

Debemos imprimir dichos valores tal como se muestra en el siguiente formato:

CANDIDATA	PUNTAJE OBTENIDO
Fernanda Torres	100
Ángela Lázaro	125
Luz Menor	90
Lucero Mendoza	45

El script para la solución del caso sería de la siguiente manera:

```
<!DOCTYPE html>
<html>
    <head>
        <meta charset="UTF-8">
        <title>Recorrer un arreglo asociativo</title>
    </head>
    <body>
        <?php
           $puntaje=array('Fernanda Torres'=>100,
                          'Ángela Lázaro'=>125,
                          'Luz Menor'=>90,
                          'Lucero Mendoza'=>45);
        ?>
        <table border="1" width="400" cellspacing="0" cellpadding="0">
            <tr>
                <td>Candidata</td>
                <td>Puntaje obtenido</td>
            </tr>
            <?php
                foreach ($puntaje as $candidata => $p) {
            ?>
            <tr>
                <td><?php echo $candidata; ?></td>
                <td><?php echo $p; ?></td>
            </tr>
            <?php
                }
            ?>
        </table>
    </body>
</html>
```

El resultado es:

Candidata	Puntaje obtenido
Fernanda Torres	100
Ángela Lázaro	125
Luz Menor	90
Lucero Mendoza	45

7.3.5 Modificar elementos

La modificación de elementos de un arreglo obedece a la demanda de cambios en algunos valores importantes del arreglo, como por ejemplo actualizar un valor de conteo o acumular montos de compras, etc. Veremos entonces cómo se modifican los valores en cada uno de los tipos de arreglos:

- **Modificar elementos en un arreglo indexado**, si tenemos el siguiente arreglo de elementos:

```
$categorias = array('Jefe', 'Operario', 'Administrativo', 'Limpieza');
```

Sólo fines educativos - FreeLibros

Si necesitamos modificar la categoría **Operario** por **Planta**, debemos ejecutar la siguiente instrucción:

```
<?php
    //Declarando e inicializando el arreglo
    $categorias = array('Jefe', 'Operario',
                        'Administrativo', 'Limpieza');

    //Modificar
    $categorias[1]='Planta';

    //Imprimir los nuevos valores
    for($i=0;$i<count($categorias);$i++){
        echo $categorias[$i].'<br>';
    }
?>
```

- **Modificar elementos en un arreglo asociativo**, si tenemos el siguiente arreglo de elementos:

```
$categorias = array('Jefe'=>3500, 'Operario'=>1200,
                    'Administrativo'=>2500, 'Limpieza'=>900);
```

Si necesitamos modificar el sueldo de la categoría **Operario** por **1500**, debemos ejecutar la siguiente instrucción:

```
<?php
    //Declarando e inicializando el arreglo
    $categorias = array('Jefe'=>3500, 'Operario'=>1200,
                        'Administrativo'=>2500, 'Limpieza'=>900);

    //Modificar
    $categorias['Operario']=1500;

    //Imprimir los nuevos valores
    foreach ($categorias as $empleado => $sueldo) {
        echo $empleado;
        echo $sueldo.'<br>';
    }
?>
```

Si necesitamos aumentar un 20 % a todos los sueldos debemos ejecutar el siguiente código:

```
<?php
    //Declarando e inicializando el arreglo
    $categorias = array('Jefe'=>3500, 'Operario'=>1200,
                        'Administrativo'=>2500, 'Limpieza'=>900);

    //Actualizando los sueldos en un 20%
    foreach ($categorias as $empleado => $sueldo) {
        $categorias[$empleado]=$sueldo*1.20;
    }

    //Imprimir los nuevos valores
    foreach ($categorias as $empleado => $sueldo) {
        echo $empleado;
        echo $sueldo.'<br>';
    }
?>
```

Sólo fines educativos - FreeLibros

7.3.6 Extrayendo un rango de elementos con array_slice

Permite obtener un bloque de elementos desde un arreglo, su formato es:

```
$nuevo_arreglo = array_slice($arreglo, posición, cantidad, true)
```

Donde:

- **$nuevo_arreglo**: Almacena en un arreglo los nuevos valores obtenidos desde la función **array_slice**.
- **$arreglo**: Es el nombre del arreglo de donde se obtendrán los valores.
- **posición**: Es la posición desde donde se obtendrán los valores a extraer.
- **cantidad**: Es el total de elementos a extraer, este valor es opcional.
- **true**: Permite conservar las llaves en el nuevo arreglo.

Si contamos con el siguiente arreglo:

```
$categorias = array('Jefe'=>3500, 'Operario'=>1200,
              'Administrativo'=>2500, 'Limpieza'=>900);
```

Y necesitamos implementar un código en PHP que permita mostrar la mitad de los elementos hasta el final, registrados en el arreglo. El código sería:

```
<?php
    //Declarando e inicializando el arreglo
      $categorias = array('Jefe'=>3500, 'Operario'=>1200,
                          'Administrativo'=>2500, 'Limpieza'=>900);

    //Extraer la mitad del arreglo
    $n=count($categorias)/2;
    $mitad=array_slice($categorias,$n);

    //Imprimir los nuevos elementos
    foreach ($mitad as $empleado => $sueldo) {
        echo $empleado;
        echo $sueldo.'<br>';
    }
?>
```

7.3.7 Avanzar y retroceder por elementos

Hay que tener en cuenta que unas colecciones de elementos, como los arreglos, tienen un puntero interno que permite ubicarse en un determinado elemento. Ahora veremos las funciones que permiten desplazarse por elementos, es decir, mover al puntero:

FUNCIÓN	DESCRIPCIÓN
next()	Permite ubicar el puntero del arreglo en una posición adelante. next($arreglo);
prev()	Permite ubicar el puntero del arreglo en una posición anterior. prev($arreglo);
end()	Permite ubicar el puntero del arreglo al final de los elementos. end($arreglo);
reset()	Permite ubicar el puntero del arreglo en el inicio de los elementos. reset($arreglo);

Sólo fines educativos - FreeLibros

También veremos algunas funciones que permiten obtener información a partir del arreglo:

FUNCIÓN	DESCRIPCIÓN
current()	Permite devolver el elemento que se encuentre en el puntero actual. current($arreglo);
pos()	Permite devolver el elemento que se encuentre en el puntero actual. pos($arreglo);
key()	Permite devolver la clave del elemento que se encuentra en el puntero actual. key($arreglo);

count()	Permite contar el total de elementos de un arreglo, también puede usar la función **sizeof()**.
	count($arreglo);
	sizeof($arreglo);

unset()	Permite destruir una variable. unset($variable);

Veamos algunos casos:

- Script que permita listar los elementos del siguiente arreglo indexado, usando la función **count** y **next**:

Libros de programación
Visual Basic 2012
PHP Fundamentos
SQL Server Administración
Programación VBA

```
<?php
     //Declaración e inicialización de elementos del arreglo
     $libros = array('Visual Basic 2012', 'PHP Fundamentos',
                     'PHP Acceso a datos',
                     'SQL Server Administración',
                     'Programación VBA' );

     //Los elementos del arreglo
     echo 'Los libros registrados son:<br>';
     for($i=0;$i<count($libros);$i++){
            echo current($libros).'<br>';
            next($libros);
     }
?>
```

O también podría implementarse usando la estructura repetitiva **do**:

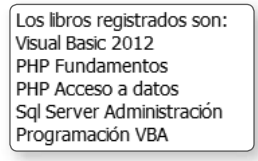
```
<?php
     //Declaración e inicialización de elementos del arreglo
     $libros = array('Visual Basic 2012', 'PHP Fundamentos',
                     'PHP Acceso a datos',
                     'SQL Server Administración',
                     'Programación VBA' );

     //Los elementos del arreglo
     echo 'Los libros registrados son:<br>';
     do{
         echo current($libros).'<br>';
     } while(next($libros));
?>
```

Sólo fines educativos - FreeLibros

O podríamos usar una función para la impresión de la siguiente manera:

```
<?php
     //Declaración e inicialización de elementos del arreglo
     $libros = array('Visual Basic 2012', 'PHP Fundamentos',
                     'PHP Acceso a datos',
                     'SQL Server Administración',
                     'Programación VBA' );

     //Los elementos del arreglo
     echo 'Los libros registrados son:<br>';
     listaLibros($libros);

     //Función de impresión de los libros
     function listaLibros($misLibros){
         do{
            echo current($misLibros).'<br>';
         }while(next($misLibros));
     }
?>
```

El resultado es:



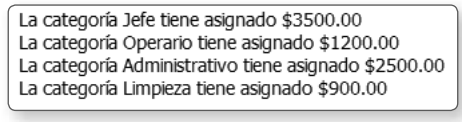
- Script que permita listar las categorías y los sueldos registrados en un arreglo asociativo, use las funciones **reset**, **sizeof**, **key**, **next** y **current**:

Categoría	**Sueldo**
Jefe	$3500.00
Operario	$1200.00
Administrativo	$2500.00
Limpieza	$900.00

```
<?php
    //Declarando e inicializando el arreglo
    $categorias = array('Jefe'=>3500, 'Operario'=>1200,
                        'Administrativo'=>2500, 'Limpieza'=>900);

    //Moviendo el puntero al inicio del arreglo
    reset($categorias);

    //Imprimiendo los valores del arreglo
    for($i=0;$i<sizeof($categorias);$i++){
        echo 'La categoría '.key($categorias).
             ' tiene asignado $'.
             number_format(current($categorias),'2','.','').
            '<br/>';
        next($categorias);
    }
?>
```

Sólo fines educativos - FreeLibros

El resultado es:

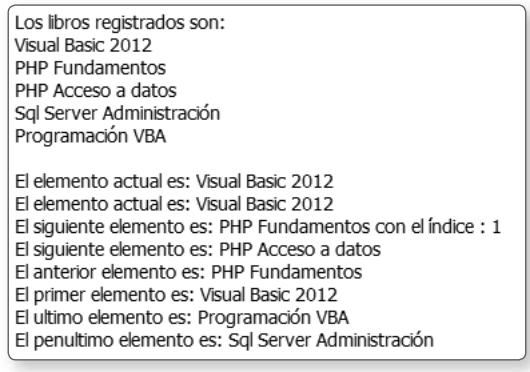


- Script que permite desplazarse por los elementos del siguiente conjunto de elementos:

Libros de programación
Visual Basic 2012
PHP Fundamentos
SQL Server Administración
Programación VBA

```
<?php
    //Declaración e inicialización de elementos del arreglo
    $libros = array('Visual Basic 2012', 'PHP Fundamentos',
                    'PHP Acceso a datos',
                    'SQL Server Administración',
                    'Programación VBA' );

    //Los elementos originales del arreglo
    echo 'Los libros registrados son:<br>';
    foreach ($libros as $descripcion)
         echo $descripcion.'<br>';

    //Aplicando funciones de desplazamiento del arreglo
    reset($libros);
    echo '<br> El elemento actual es: '.current($libros);
    echo '<br> El elemento actual es: '.pos($libros);

    echo '<br> El siguiente elemento es: '.next($libros);
    echo ' con el índice : '.key($libros);

    echo '<br> El siguiente elemento es: '.next($libros);
    echo '<br> El anterior elemento es: '.prev($libros);

    echo '<br> El primer elemento es: '.reset($libros);
    echo '<br> El ultimo elemento es: '.end($libros);
    echo '<br> El penúltimo elemento es: '.prev($libros);
?>
```

El resultado es:



Sólo fines educativos - FreeLibros

- Script que permita listar las categorías y los sueldos registrados en un arreglo asociativo usando **while**, **list** y **each**:

Categoría	Sueldo
Jefe	$3500.00
Operario	$1200.00
Administrativo	$2500.00
Limpieza	$ 900.00

```
<?php
    //Declarando e inicializando el arreglo
    $categorias = array('Jefe'=>3500, 'Operario'=>1200,
                        'Administrativo'=>2500, 'Limpieza'=>900);

    //Moviendo el puntero al inicio del arreglo
    reset($categorias);

    //Imprimiendo los valores del arreglo
    while (list($categoria, $sueldo) = each($categorias)) {
          echo 'La categoria '.$categoria.
               'tiene asignado $'.
                number_format($sueldo,'2','.','').
                '<br/>';
    }
?>
```

El resultado es:

```
La categoría Jefe tiene asignado $3500.00
La categoría Operario tiene asignado $1200.00
La categoría Administrativo tiene asignado $2500.00
La categoría Limpieza tiene asignado $900.00
```

- Script que permita listar las categorías y los sueldos registrados en un arreglo asociativo; hay que tener en cuenta que los términos «Categoría» y «Sueldo» también deben ser elementos del arreglo:

Categoría	Sueldo
Jefe	$3500.00
Operario	$1200.00
Administrativo	$2500.00
Limpieza	$ 900.00

Entonces tenemos que el primer elemento del arreglo es «Categoría Sueldo», aquí es donde usaremos la función **reset** para ubicarnos al inicio del arreglo y obtener la cabecera; luego, con una estructura repetitiva, imprimiremos los demás elementos:

```
<?php
    //Declarando e inicializando el arreglo
    $categorias = array('Categoría'=>'Sueldo','Jefe'=>3500,
                        'Operario'=>1200, Administrativo'=>2500,
                         'Limpieza'=>900);

    //Moviendo el puntero al inicio del arreglo
    reset($categorias);
```

Sólo fines educativos - FreeLibros

```
        //Obtener la cabecera
        list($categoria, $sueldo) = each($categorias);

        //Imprimiendo la cabecera
        echo '<pre>';
        echo "$categoria \t\t $sueldo <br/>";
        echo'</pre>';

        //Imprimiendo los valores del arreglo
        while (list($categoria, $sueldo) = each($categorias)) {
            echo '<pre>';
            echo "$categoria \t\t $".
                 number_format($sueldo,'2','.','')."<br/>";
            echo '</pre>';
        }
?>
```

El resultado es:

Categoría	Sueldo
Jefe	$3500.00
Operario	$1200.00
Administrativo	$2500.00
Limpieza	$900.00

7.3.8 Eliminar elementos

Eliminar un elemento del arreglo implica tener exactamente la posición del elemento, ya sea por índice o asociativo; eliminar el elemento y retroceder todos los demás elementos.

Veamos algunos casos:

- Script que permita eliminar el tercer, cuarto y quinto elemento de un arreglo indexado de productos:

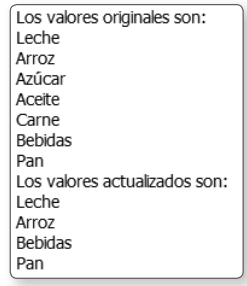
Productos
Leche
Arroz
Azúcar
Aceite
Carne
Bebidas
Pan

```
<?php
    //Declarando e inicializando el arreglo
    $productos = array('Leche','Arroz','Azúcar','Aceite',
                       'Carne','Bebidas','Pan');

    //Mostrando los valores originales
    echo 'Los valores originales son:<br>';
    foreach ($productos as $descripcion){
        echo $descripcion.'<br>';
    }
```

Sólo fines educativos - FreeLibros

```
    //Eliminando los productos
    array_splice($productos, 2, 3);

    //Mostrando los elementos actualizados
    echo 'Los valores actualizados son:<br>';
    foreach ($productos as $descripcion){
        echo $descripcion.'<br>';
    }
?>
```

El resultado es:



- Script que permita eliminar el tercer, cuarto y quinto elemento de un arreglo indexado de productos, y almacenarlo en un nuevo arreglo para luego mostrarlo en una lista:

Productos
Leche
Arroz
Azúcar
Aceite
Carne
Bebidas
Pan

```
<?php
    //Declarando e inicializando el arreglo
    $productos = array('Leche','Arroz','Azúcar','Aceite',
                       'Carne','Bebidas','Pan');

    //Mostrando los valores originales
    echo 'Los valores originales son:<br>';
    foreach ($productos as $descripcion){
        echo $descripcion.'<br>';
    }

    //Eliminando
    $eliminados = array_splice($productos, 2, 3);

    //Mostrando los valores eliminados
    echo 'Los valores eliminados son: <br>';
    foreach ($eliminados as $descripcion){
        echo $descripcion.'<br>';
    }
?>
```

Sólo fines educativos - FreeLibros

El resultado es:

```
Los valores originales son:
Leche
Arroz
Azúcar
Aceite
Carne
Bebidas
Pan
Los valores eliminados son:
Azúcar
Aceite
Carne
```

- Script que permita eliminar la categoría **Operario** de un arreglo de categorías asociativas, usando la función **unset**:

Categoría	**Sueldo**
Jefe	$3500.00
Operario	$1200.00
Administrativo	$2500.00
Limpieza	$900.00

```
<?php
    //Declarando e inicializando el arreglo
    $categorias = array('Jefe'=>3500, 'Operario'=>1200,
                        'Administrativo'=>2500, 'Limpieza'=>900);

    //Mostrando los valores originales
    echo 'Los valores originales son:<br>';
    foreach ($categorias as $cargos=>$monto){
        echo $cargos.' - '.$monto.'<br>';
    }

    //Eliminando un cargo
    unset($categorias['Operario']);

    //Mostrando los elementos actualizados
    echo '<br>Los elementos actualizados son:<br>';
    foreach ($categorias as $cargos=>$monto){
        echo $cargos.' - '.$monto.'<br>';
    }
?>
```

El resultado es:

```
Los valores originales son:
Jefe - 3500
Operario - 1200
Administrativo - 2500
Limpieza - 900

Los elementos actualizados son:
Jefe - 3500
Administrativo - 2500
Limpieza - 900
```

Sólo fines educativos - FreeLibros

7.4 MÉTODOS DE UN ARREGLO

7.4.1 Ordenamiento de elementos

Hay que tener en cuenta que unas colecciones de elementos, como son los arreglos, tienen un puntero interno que permite ubicarse en un determinado elemento. Ahora veremos las funciones que permiten desplazarse por elementos, es decir, mover al puntero:

FUNCIÓN	DESCRIPCIÓN
sort()	Permite ordenar en forma ascendente a los elementos de un arreglo indexado. sort ($arreglo);
rsort()	Permite ordenar en forma descendente a los elementos de un arreglo indexado. rsort ($arreglo);
asort()	Permite ordenar en forma ascendente los valores de un arreglo asociativo. asort ($arreglo);
arsort()	Permite ordenar en forma descendente los valores de un arreglo asociativo. arsort ($arreglo);
ksort()	Permite ordenar en forma ascendente las claves de un arreglo asociativo. ksort ($arreglo);
krsort()	Permite ordenar en forma descendente las claves de un arreglo asociativo. krsort ($arreglo);

Veamos algunos casos:

- Script que permita ordenar en forma ascendente y descendente desde un arreglo indexados de productos:

PRODUCTOS
Lavadora
Radiograbadora
Licuadora
Extractora
Afeitadora
Lámpara
Cocina
Lavavajillas
Batidora
Secadora
Tostadora
Aspiradora
Televisor
Campana
Microondas
Plancha
Calentador
Cafetera

Sólo fines educativos - FreeLibros

```php
<?php
    //Arreglo de productos
    $productos = array( 'Lavadora','Radiograbadora','Licuadora',
                        'Extractora','Afeitadora','Lámpara',
                        'Cocina','Lavavajillas','Batidora',
                        'Secadora','Tostadora','Aspiradora',
                        'Televisor','Campana','Microondas',
                        'Plancha','Calentador','Cafetera');

    //Ordenando de forma ascendente
    sort($productos);
    echo "<strong>
              Orden ascendente de los productos
          </strong><br/>";

    foreach ($productos as $i => $descripcion) {
        echo "Indice: " . $i . " -- " . $descripcion . "<br/>";
    }

    //Ordenando en forma descendente
    rsort ($productos);
    echo "<strong>
               Orden descendente de los productos
          </strong><br/>";

    foreach ($productos as $i => $descripcion) {
        echo "Indice: " . $i . " -- " . $descripcion . "<br/>";
    }
?>
```

El resultado es:

Orden ascendente de los productos
Indice: 0 -- Afeitadora
Indice: 1 -- Aspiradora
Indice: 2 -- Batidora
Indice: 3 -- Cafetera
Indice: 4 -- Calentador
Indice: 5 -- Campana
Indice: 6 -- Cocina
Indice: 7 -- Extractora
Indice: 8 -- Lámpara
Indice: 9 -- Lavadora
Indice: 10 -- Lavavajillas
Indice: 11 -- Licuadora
Indice: 12 -- Microondas
Indice: 13 -- Plancha
Indice: 14 -- Radiograbadora
Indice: 15 -- Secadora
Indice: 16 -- Televisor
Indice: 17 -- Tostadora

Orden descendente de los productos
Indice: 0 -- Tostadora
Indice: 1 -- Televisor
Indice: 2 -- Secadora
Indice: 3 -- Radiograbadora
Indice: 4 -- Plancha
Indice: 5 -- Microondas
Indice: 6 -- Licuadora
Indice: 7 -- Lavavajillas
Indice: 8 -- Lavadora
Indice: 9 -- Lámpara
Indice: 10 -- Extractora
Indice: 11 -- Cocina
Indice: 12 -- Campana
Indice: 13 -- Calentador
Indice: 14 -- Cafetera
Indice: 15 -- Batidora
Indice: 16 -- Aspiradora
Indice: 17 -- Afeitadora

Sólo fines educativos - FreeLibros

- Script que permita ordenar en forma ascendente y descendente desde un arreglo asociativo desde la descripción del producto:

PRODUCTOS	PRECIO
Lavadora	$1500.00
Radiograbadora	$500.00
Licuadora	$400.00
Extractora	$700.00
Afeitadora	$80.00
Lámpara	$50.00
Cocina	$1300.00
Lavavajillas	$170.00
Batidora	$100.00
Secadora	$1000.00
Tostadora	$60.00
Aspiradora	$250.00
Televisor	$2500.00
Campana	$700.00
Microondas	$800.00
Plancha	$150.00
Calentador	$1200.00
Cafetera	$50.00

```
<?php
   //Arreglo de productos
   $productos = array( 'Lavadora'=>1500,'Radiograbadora'=>500,
                       'Licuadora'=>400,'Extractora'=>700,
                       'Afeitadora'=>80,'Lampara'=>50,
                       'Cocina'=>1300,'Lavavajillas'=>170,
                       'Batidora'=>100,'Secadora'=>1000,
                       'Tostadora'=>60,'Aspiradora'=>250,
                       'Televisor'=>2500,'Campana'=>700,
                       'Microondas'=>800,'Plancha'=>150,
                       'Calentador'=>1200,'Cafetera'=>50);

   //Ordenando en forma ascendente por la descripcion
   ksort ($productos) ;
   echo "<strong>
          Orden ascendente según la descripcion:
         </strong><br/>";
   foreach ($productos as $descripcion => $precio) {
        echo $descripcion . " -- " . $precio . "<br/>";
   }

   //Ordenando en forma descendente por la descripcion
   krsort ($productos) ;
   echo "<strong>
           Orden descendente según la descripcion:
         </strong><br/>";
   foreach ($productos as $descripcion => $precio) {
        echo $descripcion . " -- " . $precio . "<br/>";
   }
?>
```

Sólo fines educativos - FreeLibros

El resultado es:

Orden ascendente segun la descripcion:
Afeitadora -- 80
Aspiradora -- 250
Batidora -- 100
Cafetera -- 50
Calentador -- 1200
Campana -- 700
Cocina -- 1300
Extractora -- 700
Lámpara -- 50
Lavadora -- 1500
Lavavajillas -- 170
Licuadora -- 400
Microondas -- 800
Plancha -- 150
Radiograbadora -- 500
Secadora -- 1000
Televisor -- 2500
Tostadora -- 60

Orden descendente segun la descripcion:
Tostadora -- 60
Televisor -- 2500
Secadora -- 1000
Radiograbadora -- 500
Plancha -- 150
Microondas -- 800
Licuadora -- 400
Lavavajillas -- 170
Lavadora -- 1500
Lámpara -- 50
Extractora -- 700
Cocina -- 1300
Campana -- 700
Calentador -- 1200
Cafetera -- 50
Batidora -- 100
Aspiradora -- 250
Afeitadora -- 80

- Script que permita ordenar en forma ascendente y descendente desde un arreglo asociativo a partir el precio del producto:

PRODUCTOS	PRECIO
Lavadora	$1500.00
Radiograbadora	$500.00
Licuadora	$400.00
Extractora	$700.00
Afeitadora	$80.00
Lámpara	$50.00
Cocina	$1300.00
Lavavajillas	$170.00
Batidora	$100.00
Secadora	$1000.00
Tostadora	$60.00
Aspiradora	$250.00
Televisor	$2500.00
Campana	$700.00
Microondas	$800.00
Plancha	$150.00
Calentador	$1200.00
Cafetera	$50.00

```php
<?php
   //Arreglo de productos
   $productos = array( 'Lavadora'=>1500,'Radiograbadora'=>500,
                       'Licuadora'=>400,'Extractora'=>700,
                       'Afeitadora'=>80,'Lampara'=>50,
                       'Cocina'=>1300,'Lavavajillas'=>170,
                       'Batidora'=>100,'Secadora'=>1000,
                       'Tostadora'=>60,'Aspiradora'=>250,
                       'Televisor'=>2500,'Campana'=>700,
                       'Microondas'=>800,'Plancha'=>150,
                       'Calentador'=>1200,'Cafetera'=>50);

   //Ordenado en forma ascendente por el precio
   asort ($productos);
   echo "<strong>Orden ascendente según el precio:</strong><br/>";
   foreach ($productos as $descripcion => $precio) {
        echo $descripcion. " -- " . $precio. "<br/>";
   }
```

Sólo fines educativos - FreeLibros

```
    //Ordenando en forma descendente por el precio
    arsort ($productos);
    echo "<strong>
                Orden descendente según el precio:
           </strong><br/>";
    foreach ($productos as $descripcion => $precio) {
         echo $descripcion . " -- " .$precio. "<br/>";
    }
?>
```

El resultado es:

Orden descendente según el precio:
Televisor -- 2500
Lavadora -- 1500
Cocina -- 1300
Calentador -- 1200
Secadora -- 1000
Microondas -- 800
Campana -- 700
Extractora -- 700
Radiograbadora -- 500
Licuadora -- 400
Aspiradora -- 250
Lavavajillas -- 170
Plancha -- 150
Batidora -- 100
Afeitadora -- 80
Tostadora -- 60
Cafetera -- 50
Lámpara -- 50

Orden ascendente según el precio:
Lámpara -- 50
Cafetera -- 50
Tostadora -- 60
Afeitadora -- 80
Batidora -- 100
Plancha -- 150
Lavavajillas -- 170
Aspiradora -- 250
Licuadora -- 400
Radiograbadora -- 500
Campana -- 700
Extractora -- 700
Microondas -- 800
Secadora -- 1000
Calentador -- 1200
Cocina -- 1300
Lavadora -- 1500
Televisor -- 2500

7.4.2 Convertir un arreglo en una lista de variables

- **Función extract**: Es una función que permite convertir en variables todas las claves de un arreglo asociativo, de tal manera que es asignado con el valor determinado en la inicialización del arreglo. Su formato es:

```
extract($arreglo);
```

Veamos lo que ocurre si se necesita imprimir los precios de los productos lavadora y plancha desde la siguiente lista de productos:

PRODUCTOS	PRECIO
Lavadora	$1500.00
Radiograbadora	$500.00
Licuadora	$400.00
Extractora	$700.00
Afeitadora	$80.00
Lámpara	$50.00
Cocina	$1300.00
Lavavajillas	$170.00
Batidora	$100.00
Secadora	$1000.00
Tostadora	$60.00
Aspiradora	$250.00
Televisor	$2500.00
Campana	$700.00
Microondas	$800.00
Plancha	$150.00
Calentador	$1200.00
Cafetera	$50.00

Sólo fines educativos - FreeLibros

```php
<?php
   //Arreglo de productos
   $productos = array( 'Lavadora'=>1500,'Radiograbadora'=>500,
                       'Licuadora'=>400,'Extractora'=>700,
                       'Afeitadora'=>80,'Lampara'=>50,
                       'Cocina'=>1300,'Lavavajillas'=>170,
                       'Batidora'=>100,'Secadora'=>1000,
                       'Tostadora'=>60,'Aspiradora'=>250,
                       'Televisor'=>2500,'Campana'=>700,
                       'Microondas'=>800,'Plancha'=>150,
                       'Calentador'=>1200,'Cafetera'=>50);

  //Convirtiendo el arreglo en variables
  extract($productos);

  echo 'El precio de la Lavadora es: '.$Lavadora;
  echo '<br>El precio de la Plancha es: '.$Plancha;
?>
```

El resultado es:

```
El precio de la Lavadora es: 1500
El precio de la Plancha es: 150
```

Veamos el caso en el cual se impriman los precios de los productos lavadora y refrigeradora desde la siguiente lista de productos:

PRODUCTOS	PRECIO
Lavadora	$1500.00
Radiograbadora	$500.00

```php
<?php
    //Declarando e inicializando el arreglo
    $productos = array('Lavadora' => 1500,
                       'Refrigeradora' => 3500);

    //Extrayendo los productos en la variable prefijada
    extract($productos, EXTR_PREFIX_ALL, 'producto');

    //Mostrando los precios
    echo 'El precio de la Lavadora es: $'.
                number_format($producto_Lavadora,'2','.','');
    echo '<br>El precio de la Refrigeradora es: $'.
                number_format($producto_Refrigeradora,'2','.','');
?>
```

El resultado es:

```
El precio de la Lavadora es: $1500.00
El precio de la Refrigeradora es: $3500.00
```

- **Función list**: Asigna a variables los elementos de un arreglo, tanto la clave como el valor.

```php
list($variable1, $variable2) = each($arreglo);
```

Sólo fines educativos - FreeLibros

Veamos el script que permite listar los productos usando la función **list** desde el arreglo indexado productos:

PRODUCTOS
Lavadora
Radiograbadora
Licuadora
Extractora
Afeitadora
Lámpara
Cocina
Lavavajillas
Batidora

PRODUCTOS
Secadora
Tostadora
Aspiradora
Televisor
Campana
Microondas
Plancha
Calentador
Cafetera

```
<?php
    //Arreglo de productos
    $productos = array( 'Lavadora','Radiograbadora','Licuadora',
                        'Extractora','Afeitadora','Lámpara',
                        'Cocina','Lavavajillas','Batidora',
                        'Secadora','Tostadora','Aspiradora',
                        'Televisor','Campana','Microondas',
                        'Plancha','Calentador','Cafetera');

    //Listando los productos
    echo 'Los Productos registrados son: <br>';
    for($i=0;$i<count($productos);$i++){
        //Obtener información desde el arreglo productos
        list(,$descripcion)=each($productos);
        echo $descripcion.'<br>';
    }
?>
```

También podríamos usar **while** para el listado de productos de la siguiente manera:

```
<?php
    //Arreglo de productos
    $productos = array( 'Lavadora','Radiograbadora','Licuadora',
                        'Extractora','Afeitadora','Lámpara',
                        'Cocina','Lavavajillas','Batidora',
                        'Secadora','Tostadora','Aspiradora',
                        'Televisor','Campana','Microondas',
                        'Plancha','Calentador','Cafetera');

    //Listando los productos
    echo 'Los productos son: <br>';
    while(list(,$descripcion) = each($productos)){
        echo $descripcion.'<br>';
    }
?>
```

Sólo fines educativos - FreeLibros

El resultado es:

Finalmente, veamos el caso donde se necesite imprimir tanto la descripción de los productos como el precio del mismo por medio de la función **list**:

PRODUCTOS	PRECIO
Lavadora	$1500.00
Radiograbadora	$500.00
Licuadora	$400.00
Extractora	$700.00
Afeitadora	$80.00
Lámpara	$50.00
Cocina	$1300.00
Lavavajillas	$170.00
Batidora	$100.00
Secadora	$1000.00
Tostadora	$60.00
Aspiradora	$250.00
Televisor	$2500.00
Campana	$700.00
Microondas	$800.00
Plancha	$150.00
Calentador	$1200.00
Cafetera	$50.00

```
<?php
  //Arreglo de productos
   $productos = array( 'Lavadora'=>1500,'Radiograbadora'=>500,
                       'Licuadora'=>400,'Extractora'=>700,
                       'Afeitadora'=>80,'Lampara'=>50,
                       'Cocina'=>1300,'Lavavajillas'=>170,
                       'Batidora'=>100,'Secadora'=>1000,
                       'Tostadora'=>60,'Aspiradora'=>250,
                       'Televisor'=>2500,'Campana'=>700,
                       'Microondas'=>800,'Plancha'=>150,
                       'Calentador'=>1200,'Cafetera'=>50);

   //Imprimir el listado de productos
   echo '<pre>';
   echo "Descripción\t\tPrecio<br>";
   for($i=0;$i<count($productos);$i++){
            list($descripcion,$precio)=each($productos);
            echo $descripcion."\t\t $".
                 number_format($precio,'2','.','').'<br>';
   }
   echo '</pre>';
?>
```

Sólo fines educativos - FreeLibros

El resultado es:

```
Descripción             Precio
Lavadora                 $1500.00
Radiograbadora           $500.00
Licuadora                $400.00
Extractora               $700.00
Afeitadora               $80.00
Lámpara          $50.00
Cocina           $1300.00
Lavavajillas             $170.00
Batidora                 $100.00
Secadora                 $1000.00
Tostadora                $60.00
Aspiradora               $250.00
Televisor                $2500.00
Campana          $700.00
Microondas               $800.00
Plancha          $150.00
Calentador               $1200.00
Cafetera                 $50.00
```

7.4.3 Convertir cadena de caracteres en array

Para convertir una cadena de caracteres en un arreglo, la cadena debe tener un factor de separación entre sus elementos, que podría ser un espacio en blanco, una barra vertical o algo que lo diferencie entre los elementos, por ejemplo:

1,2,3,4,5,6	El factor de separación son las comas.
A B C D E	El factor de separación es el espacio en blanco.
12/10/2015|01/09/2015|19/12/2015	El factor de separación es la barra vertical **|**.

Entonces, el formato para convertir cadena de caracteres en **array** es:

```
$arreglo = explode('Factor de separación',$cadena);
```

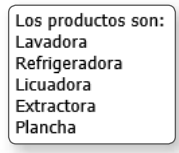
Veamos el caso donde una variable llamada **misproductos** tiene la siguiente cadena:

```
$misproductos = 'Lavadora Refrigeradora Licuadora Extractora Plancha';
```

```
<?php
    //Declaración y asignación de la variable misproductos
    $misproductos='Lavadora Refrigeradora Licuadora
                   Extractora Plancha';

    //Creando y asignando el arreglo con los elementos de misproductos
    $productos = explode(' ', $misproductos);

    //Listando los productos
    echo 'Los productos son: <br>';
    foreach ($productos as $p){
        echo $p.'<br>';
    }
?>
```

Sólo fines educativos - FreeLibros

El resultado es:



7.4.4 Eliminando elementos repetidos en un array

La idea principal es crear un arreglo con elementos que no sean repetidos, algo así como la moda estadística, en la cual los elementos repetidos serán omitidos de la lista, su formato es:

```
                join('Factor de cambio',array_unique($arreglo));
               implode('Factor de cambio',array_unique($arreglo));
```

Veamos el caso en que una variable de arreglo **$paises** tiene algunos países repetidos en su registro, como se muestra a continuación:

```
$paises = array('Perú','Ecuador','Argentina','Uruguay',
                'Colombia','Bolivia','Colombia','Santo Domingo',
                'Venezuela','Argentina','Brasil','Perú');
```

Listaremos los mismos países omitiendo los repetidos de la siguiente manera:

```
<?php
    //Arreglo de paises
    $paises = array('Perú','Ecuador','Argentina','Uruguay',
                    'Colombia','Bolivia','Colombia','Santo Domingo',
                    'Venezuela','Argentina','Brasil','Perú');

    //Creando el arreglo paisesS omitiendo los repetidos
    $paisesS = array_unique($paises);

    //Listando los nuevos elementos
    echo 'Los paises son: <br>';
    foreach ($paisesS as $p){
        echo $p.'<br>';
    }
?>
```

El resultado es:

Sólo fines educativos - FreeLibros

7.4.5 Invertir los elementos de un arreglo

La inversión de los elementos de un arreglo trata de invertir una clave por un valor y viceversa, tenemos el siguiente formato:

```
array_flip($arreglo);
```

Si contamos con un arreglo de productos y sus precios, y necesitamos invertir la clave por el valor usando la función **array_flip**:

PRODUCTOS	PRECIO
Lavadora	$1500.00
Radiograbadora	$500.00
Licuadora	$400.00
Extractora	$700.00
Afeitadora	$80.00
Lámpara	$50.00
Cocina	$1300.00
Lavavajillas	$170.00
Batidora	$100.00
Secadora	$1000.00
Tostadora	$60.00
Aspiradora	$250.00
Televisor	$2500.00
Campana	$700.00
Microondas	$800.00
Plancha	$150.00
Calentador	$1200.00
Cafetera	$50.00

```
<?php
  //Arreglo de productos
   $productos = array( 'Lavadora'=>1500,'Radiograbadora'=>500,
                       'Licuadora'=>400,'Extractora'=>700,
                       'Afeitadora'=>80,'Lampara'=>50,
                       'Cocina'=>1300,'Lavavajillas'=>170,
                       'Batidora'=>100,'Secadora'=>1000,
                       'Tostadora'=>60,'Aspiradora'=>250,
                       'Televisor'=>2500,'Campana'=>700,
                       'Microondas'=>800,'Plancha'=>150,
                       'Calentador'=>1200,'Cafetera'=>50);

   //Invirtiendo los elementos del arreglo
   $productosI = array_flip($productos);

   //Listado de los productos invertidos
   echo '<pre>';
   echo "Descripción\t\tPrecio<br>";
   for($i=0;$i<count($productosI);$i++){
            list($precio,$descripcion)=each($productosI);
            echo $descripcion."\t\t $".
                 number_format($precio,'2','.','').'<br>';
   }
   echo '</pre>';
?>
```

Sólo fines educativos - FreeLibros

El resultado es:

```
Descripción              Precio
Lavadora                  $1500.00
Radiograbadora            $500.00
Licuadora                 $400.00
Extractora                $700.00
Afeitadora                $80.00
Lámpara          $50.00
Cocina           $1300.00
Lavavajillas              $170.00
Batidora                  $100.00
Secadora                  $1000.00
Tostadora                 $60.00
Aspiradora                $250.00
Televisor                 $2500.00
Campana          $700.00
Microondas                $800.00
Plancha          $150.00
Calentador                $1200.00
Cafetera                  $50.00
```

7.5 Arreglos multidimensionales

Se le llama así cuando un arreglo cuenta con más de un índice, esto ocurre cuando un elemento de un arreglo unidimensional contiene muchos valores más, es decir, otro arreglo. Por ejemplo:

EMP001	EMP002	EMP003	EMP004
Juan Pérez	María López	Carlos Caraza	Martín Martínez
2000.00	1400.00	2000.00	800.00
10/10/2010	10/02/2010	21/12/2011	12/08/2011
A	B	A	D

Veamos el código que implementa el arreglo multidimensional de los datos de los empleados:

```
<?php
    //Declaración e inicialización de elementos del arreglo
    $empleado = array('EMP001','EMP002','EMP003','EMP004');

    $empleado['EMP001']=array('Nombre'=>'Juan Pérez',
                              'Sueldo'=>2000,
                              'FechaReg'=>'10/10/2010',
                              'Categoría'=>'A');

    $empleado['EMP002']=array('Nombre'=>'María López',
                              'Sueldo'=>1400,
                              'FechaReg'=>'10/02/2010',
                              'Categoría'=>'B');

    $empleado['EMP003']=array('Nombre'=>'Carlos Caraza',
                              'Sueldo'=>2000,
                              'FechaReg'=>'21/12/2011',
                              'Categoría'=>'A');

    $empleado['EMP004']=array('Nombre'=>'Martin Martínez',
                              'Sueldo'=>800,
                              'FechaReg'=>'12/08/2011',
                              'Categoría'=>'D');

    //Imprimir los datos del Empleado con código EMP001
    echo '<strong>Los datos son:</strong><br>';
    echo 'Empleado: '.$empleado['EMP001']['Nombre'].'<br>';
```

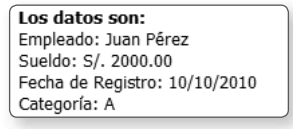
Sólo fines educativos - FreeLibros

```
    echo 'Sueldo: S/.'
         .number_format($empleado['EMP001']['Sueldo'],'2','.','')
         .'<br>';
    echo 'Fecha de Registro:'.$empleado['EMP001']['FechaReg'].'<br>';
    echo 'Categoría: '.$empleado['EMP001']['Categoría'];
?>
```

Otra forma de implementar al arreglo podría ser de la siguiente manera:

```
<?php
    //Declaración e inicialización de elementos del arreglo
    $empleados = array('EMP001'=>array('Nombre'=>'Juan Pérez',
                                'Sueldo'=>2000,
                                'FechaReg'=>'10/10/2010',
                                'Categoría'=>'A'),
                       'EMP002'=>array('Nombre'=>'María López',
                                'Sueldo'=>1400,
                                'FechaReg'=>'10/02/2010',
                                'Categoría'=>'B'),
                       'EMP003'=>array('Nombre'=>'Carlos Caraza',
                                'Sueldo'=>2000,
                                'FechaReg'=>'21/12/2011',
                                'Categoría'=>'A'),
                       'EMP004'=>array('Nombre'=>'Martin Martinez',
                                'Sueldo'=>800,
                                'FechaReg'=>'12/08/2011',
                                'Categoría'=>'D'));

    //Imprimir los datos del Empleado con código EMP001
    echo '<strong>Los datos son:</strong><br>';
    echo 'Empleado: '.$empleados['EMP001']['Nombre'].'<br>';
    echo 'Sueldo: S/. '
         .number_format($empleados['EMP001']['Sueldo'],'2','.','')
         .'<br>';
    echo 'Fecha de Registro: '.$empleados['EMP001']['FechaReg'].'<br>';
    echo 'Categoría: '.$empleados['EMP001']['Categoría'];
?>
```

El resultado es:



Para imprimir todos los elementos usaríamos el siguiente código:

```
<?php
//Declaración e inicialización de elementos del arreglo
    $empleados = array('EMP001'=>array('Nombre'=>'Juan Pérez',
                                'Sueldo'=>2000,
                                'FechaReg'=>'10/10/2010',
                                'Categoría'=>'A'),
                       'EMP002'=>array('Nombre'=>'María López',
                                'Sueldo'=>1400,
                                'FechaReg'=>'10/02/2010',
                                'Categoría'=>'B'),
                       'EMP003'=>array('Nombre'=>'Carlos Caraza',
                                'Sueldo'=>2000,
                                'FechaReg'=>'21/12/2011',
```

Sólo fines educativos - FreeLibros

```
                                'Categoría'=>'A'),
                        'EMP004'=>array('Nombre'=>'Martin Martinez',
                                'Sueldo'=>800,
                                'FechaReg'=>'12/08/2011',
                                'Categoría'=>'D'));

    //Imprimir todos los empleados
    echo 'LISTADO DE EMPLEADOS<br><br>';
    echo '<table border="1" width="700" cellspacing="0" cellpadding="0">';
    echo '<tr>
            <td width="100"><strong>Codigo</strong></td>
            <td width="200"><strong>Empleado</strong></td>
            <td width="100"><strong>Sueldo</strong></td>
            <td width="100"><strong>Fecha de Registro</strong></td>
            <td width="100"><strong>Categoria</strong></td>
          </tr>';

    //Imprimiendo la lista de Empleados embebido con HTML
    while(list($codigo,$datos) =  each($empleados)){
        echo '<tr><td>';
        echo $código;
        echo '</td>';
        echo '<td>';
        echo $datos['Nombre'];
        echo '</td>';
        echo '<td>';
        echo 'S/. '.number_format($datos['Sueldo'],'2','.','');
        echo '</td>';
        echo '<td>';
        echo $datos['FechaReg'];
        echo '</td>';
        echo '<td>';
        echo $datos['Categoría'];
        echo '</td></tr>';
    }
    echo '</table>';
?>
```

El resultado es:

LISTADO DE EMPLEADOS

Código	Empleado	Sueldo	Fecha de registro	Categoría
EMP001	Juan Pérez	S/. 2000.00	10/10/2010	A
EMP002	María López	S/. 1400.00	10/02/2010	B
EMP003	Carlos Caraza	S/. 2000.00	21/12/2011	A
EMP004	Martín Martínez	S/. 800.00	12/08/2011	D

Sólo fines educativos - FreeLibros

7.6 CASOS DESARROLLADOS

Para todos los casos que desarrollaremos en este capítulo, usaremos el siguiente CSS llámelo **estilo.css**:

```
body{
    font-family: tahoma;
    font-size: 14px;
}
#centrado{
    text-align: center;
}
table {
    margin: auto;
}
img{
    margin: auto;
    display: block;
}
#fila{
    font-size: 13px;
    background: #b9c9fe;
    color: #039;
}
#error{
    color: red;
    font-size: 12px;
}
#codigo{
    color: blue;
    font-size: 36px;
}
table th {
   background-color: #b9c9fe;
}
```

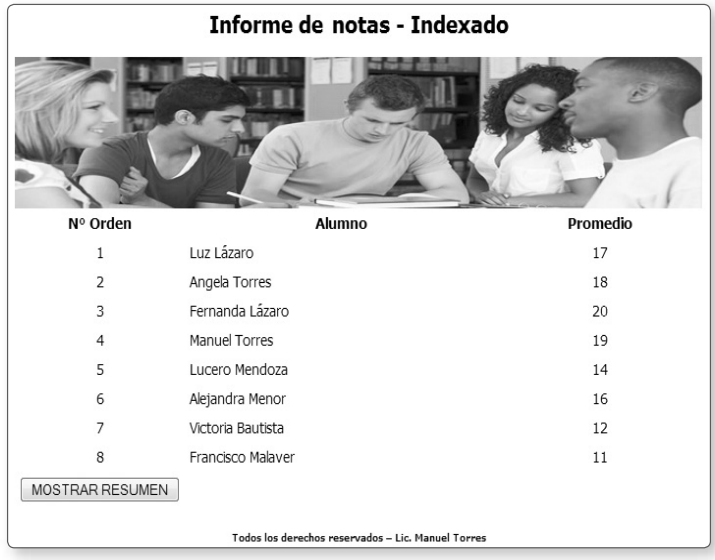
- **Caso desarrollado 1: Arreglo indexado – Informe de notas**

  Implemente una aplicación web con PHP que permita generar un informe de notas según la siguiente tabla:

NOMBRES Y APELLIDOS	PROMEDIO
Luz Lázaro	17
Ángela Torres	18
Fernanda Lázaro	20
Manuel Torres	19
Lucero Mendoza	14
Alejandra Menor	16
Victoria Bautista	12
Francisco Malaver	11

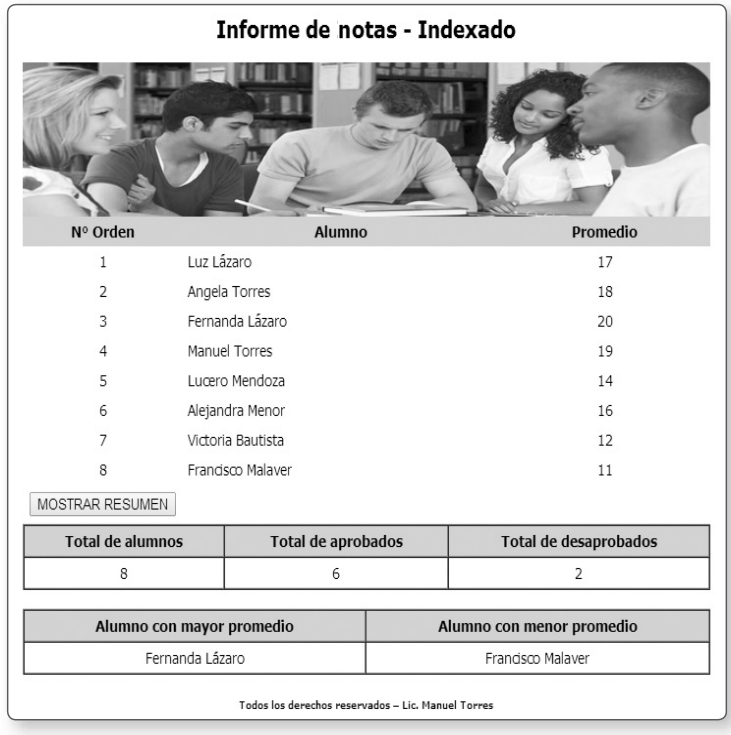
Sólo fines educativos - FreeLibros

La interfaz gráfica inicial es la siguiente:



Tener en cuenta:

- Al presionar el botón **Mostrar resumen** se mostrará el **Total de alumnos registrados**, **Total de alumnos aprobados**, **Total de alumnos desaprobados**, nombre del alumno con mayor promedio y nombre del alumno con menor promedio.
- Use funciones implementadas por el usuario para todos los procesos de cálculo.
- La presentación final de la aplicación es de la siguiente manera:



Sólo fines educativos - FreeLibros

Archivo: **notas_i.php**

```
<!DOCTYPE html>
<html>
    <head>
        <meta charset="UTF-8">
        <title>Informe de Notas</title>
        <link href="estilo.css" rel="stylesheet">
    </head>
    <body>
        <?php
            //Definición de los arreglos
            $alumnos = getAlumnos();
            $promedios = getPromedios();
        ?>
        <header>
            <h2 id="centrado">Informe de Notas - Indexado</h2>
            <img src="bannerEstudiantes.jpg" width="700" height="140"/>
        </header>
        <section>
          <form action="notas_i.php" method="POST">
            <table border="0" width="700" cellspacing="0" cellpadding="5">
                <tr>
                    <th width="10">N° Orden</th>
                    <th>Alumno</th>
                    <th>Promedio</th>
                </tr>
                <?php
                    //Imprimir
                    for($i=0;$i<getTotal();$i++){
                ?>
                <tr>
                    <td id="centrado"><?php echo $i+1; ?></td>
                    <td><?php echo $alumnos[$i]; ?></td>
                    <td id="centrado"><?php echo $promedios[$i]; ?></td>
                </tr>
                    <?php } ?>
                <tr>
                    <td><input type="submit" value="MOSTRAR RESUMEN"
                               name="btnMostrar"/>
                    </td>
                    <td></td>
                    <td></td>
                </tr>
            </table>
         </form>
            <?php
                //Total de aprobados y desaprobados
               list($tAprobados,$tDesaprobados)=totalAprobados_Desaprobados();

               //Condicionar la muestra de los resultados
               if(isset($_POST["btnMostrar"])){
            ?>
            <table border="1" width="700" cellspacing="0" cellpadding="5">
                <tr>
                    <th>Total de alumnos</th>
                    <th>Total de aprobados</th>
                    <th>Total de desaprobados</th>
                </tr>
                <tr>
                    <td id="centrado"><?php echo getTotal(); ?></td>
                    <td id="centrado"><?php echo $tAprobados; ?></td>
```

Sólo fines educativos - FreeLibros

```
                    <td id="centrado"><?php echo $tDesaprobados; ?></td>
                </tr>
            </table>
            <?php
                //Obtener el mayor y menor elemento
                list($maximo,$minimo) = valor_maximo_minimo();
                //Obtener el índice del mayor y menor elemento
                list($maIndice,$miIndice) = indice_maximo_minimo();
            ?>
            <br />
            <table border="1" width="700" cellspacing="0" cellpadding="5">
                <tr>
                    <th>Alumno con mayor promedio</th>
                    <th>Alumno con menor promedio</th>
                </tr>
                <tr>
                    <td id="centrado"><?php echo getAlumnos()[$maIndice].
                                                '('.$maximo.')'; ?></td>
                    <td id="centrado"><?php echo getAlumnos()[$miIndice].
                                                '('.$minimo.')'; ?></td>
                </tr>
            </table>
                <?php } ?>
        </section>
        <footer>
            <h6 id="centrado">Todos los derechos reservados –
                Lic. Manuel Torres</h6>
        </footer>
    </body>
</html>
<?php
    //Función de implementación para el arreglo indexado de alumnos
    function getAlumnos() {
        return array('Luz Lázaro','Ángela Torres',
                     'Fernanda Lázaro','Manuel Torres',
                     'Lucero Mendoza','Alejandra Menor',
                     'Victoria Bautista','Francisco Malaver');
    }

    //Función de implementación para el arreglo de notas
    function getPromedios() {
        return array(17,18,20,19,14,16,12,11);
    }

    //Función que determina el total de alumnos
    function getTotal(){
        return count(getAlumnos());
    }

    //Función que determina el total de aprobados y desaprobados
    function totalAprobados_Desaprobados(){
        $tAprobados = 0;
        $tDesaprobados = 0;
        for($i=0;$i<getTotal();$i++){
            if (getPromedios()[$i]<13)
                $tDesaprobados++;
                else
                $tAprobados++;
        }
        return array($tAprobados,$tDesaprobados);
    }
```

Sólo fines educativos - FreeLibros

```
    //Determinar el maximo y menor promedio
    function valor_maximo_minimo(){
        $maximo = max(getPromedios());
        $minimo = min(getPromedios());
        return array($maximo,$minimo);
    }

    //Determinar el índice del mayor y menor promedio
    function indice_maximo_minimo(){
        list($maximo,$minimo)=  valor_maximo_minimo();
        $maIndice = array_search($maximo, getPromedios());
        $miIndice = array_search($minimo, getPromedios());
        return array($maIndice,$miIndice);
    }
?>
```

Comentarios:

```
function getAlumnos() {
    return array('Luz Lázaro','Ángela Torres',
              'Fernanda Lázaro','Manuel Torres',
              'Lucero Mendoza','Alejandra Menor',
              'Victoria Bautista','Francisco Malaver');
}
```

La función **getAlumnos** permite llenar el arreglo con los nombres de los alumnos. Debe tener en cuenta que esta función retorna una matriz de alumnos, y que el primer alumno se almacena en la posición cero del arreglo.

```
function getPromedios() {
    return array(17,18,20,19,14,16,12,11);
}
```

La función **getPromedio** permite llenar el arreglo con los promedios de cada alumno, debe tener en cuenta que esta función retorna una matriz de promedios, y que el primer promedio se almacena en la posición cero del arreglo, el cual también corresponde al primer alumno. Por lo tanto, Luz Lázaro tendría registrado su promedio 17, ambos en el índice cero de sus respectivos arreglos.

```
function getTotal(){
     return count(getAlumnos());
}
```

La función **getTotal** determina el número total de alumnos registrados en el arreglo.

```
function totalAprobados_Desaprobados(){
     $tAprobados = 0;
     $tDesaprobados = 0;
     for($i=0;$i<getTotal();$i++){
        if (getPromedios()[$i]<13)
            $tDesaprobados++;
            else
            $tAprobados++;
     }
     return array($tAprobados,$tDesaprobados);
}
```

Sólo fines educativos - FreeLibros

La función **totalAprobados_Desaprobados** tiene por misión determinar el total de alumnos aprobados y desaprobados a partir de sus promedios registrados en el arreglo de promedios. Hay que tener en cuenta que debemos inicializar los contadores **$tAprobados** y **$tDesaprobados**, y que para comprobar cada promedio necesitamos una estructura repetitiva que recorra por todos los promedios. Para evaluar cada promedio debemos invocar la función **getPromedios** de modo que **getPromedios()[$i]** acceda a cada promedio. Finalmente, se retorna al total de aprobados y desaprobados mediante un arreglo; es decir, esta función devuelve dos valores al mismo tiempo.

```
function valor_maximo_minimo(){
     $maximo = max(getPromedios());
     $minimo = min(getPromedios());
     return array($maximo,$minimo);
  }
```

La función **valor_maximo_minimo** determina solo el mayor y menor promedio registrado; finalmente, ambos valores son devueltos por la función en forma de arreglo.

```
function indice_maximo_minimo(){
     list($maximo,$minimo)=  valor_maximo_minimo();
     $maIndice = array_search($maximo, getPromedios());
     $miIndice = array_search($minimo, getPromedios());
     return array($maIndice,$miIndice);
  }
```

La función **indice_maximo_minimo** tiene la misión de determinar en qué posición del arreglo se encuentran el mayor y menor valor promedio encontrado en la función **valor_maximo_minimo**.

- **list($maximo,$minimo) = valor_maximo_minimo();**
  Debemos considerar que la función **valor_maximo_minimo** devuelve un arreglo con dos elementos y estos serán enviados a las variables **$maximo** y **$minimo** por medio de la función **list**.
- **$maIndice = array_search($maximo, getPromedios());**
  La función **array_search** permite determinar la posición del elemento **$maximo** dentro del arreglo Promedio obtenido desde la función **getPromedios()**, esta información es registrada en **$maIndice**; ello nos servirá para indicar quién es el alumno con el mayor promedio.
- **$miIndice = array_search($minimo, getPromedios());**
  Trabaja de la misma manera que **$maIndice** con la diferencia de que almacena el índice del menor promedio encontrado con la función **array_search**.

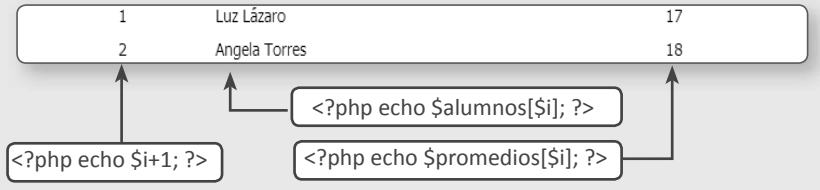
```
<?php
          $alumnos = getAlumnos();
          $promedios = getPromedios();
?>
```

$alumnos permite crear un arreglo en base a la función **getAlumnos()**, ya que esta devuelve una matriz de elementos de alumnos. De la misma manera para la variable **$promedios**.

Sólo fines educativos - FreeLibros

```
<?php
       for($i=0;$i<getTotal();$i++){
?>
<tr>
      <td id="centrado"><?php echo $i+1; ?></td>
      <td><?php echo $alumnos[$i]; ?></td>
      <td id="centrado"><?php echo $promedios[$i]; ?></td>
</tr>
<?php } ?>
```

En el siguiente script, la estructura **for** recorre desde la posición cero del arreglo hasta el último elemento encontrado, tanto en los alumnos como en los promedios. Considere que dicho **for** está encerrando a las etiquetas <tr></tr>; eso quiere decir que también construirá dinámicamente filas de una tabla para mostrar dentro de él a los alumnos con sus promedios.




```
<?php
     list($tAprobados,$tDesaprobados)=totalAprobados_Desaprobados();
     if(isset($_POST["btnMostrar"])){
?>
<table border="1" width="700" cellspacing="0" cellpadding="5">
…
```

La función **totalAprobados_Desaprobados** devuelve una matriz de dos elementos que son descargados en las variables **tAprobados** y **tDesaprobados** para poder imprimirlos más adelante. La función **if** comprueba que se haya seleccionado el botón **Mostrar** para mostrar los resultados esperados por el usuario.

```
<tr>
     <td id="centrado"><?php echo getTotal(); ?></td>
     <td id="centrado"><?php echo $tAprobados; ?></td>
     <td id="centrado"><?php echo $tDesaprobados; ?></td>
</tr>
```

Se imprime el total de alumnos desde la función **getTotal**, el total de aprobados y total de desaprobados descargados desde la función **totalAprobados_Desaprobados**.

```
<?php
      list($maximo,$minimo) = valor_maximo_minimo();
      list($maIndice,$miIndice) = indice_maximo_minimo();
?>
```

Se obtiene el máximo-menor promedio y son almacenados en las variables **$maximo** y **$minimo**; de la misma manera para el índice de ambos promedios. Recuerde que las funciónes **valor_maximo_minimo** e **índice_maximo_minimo** retornar un **array** que puede ser asignado a variables comunes con la función **list**.

Sólo fines educativos - FreeLibros

```
<tr>
  <td id="centrado"><?php echo getAlumnos()[$maIndice]. '('.$maximo.')'; ?></td>
  <td id="centrado"><?php echo getAlumnos()[$miIndice]. '('.$minimo.')'; ?></td>
</tr>
```

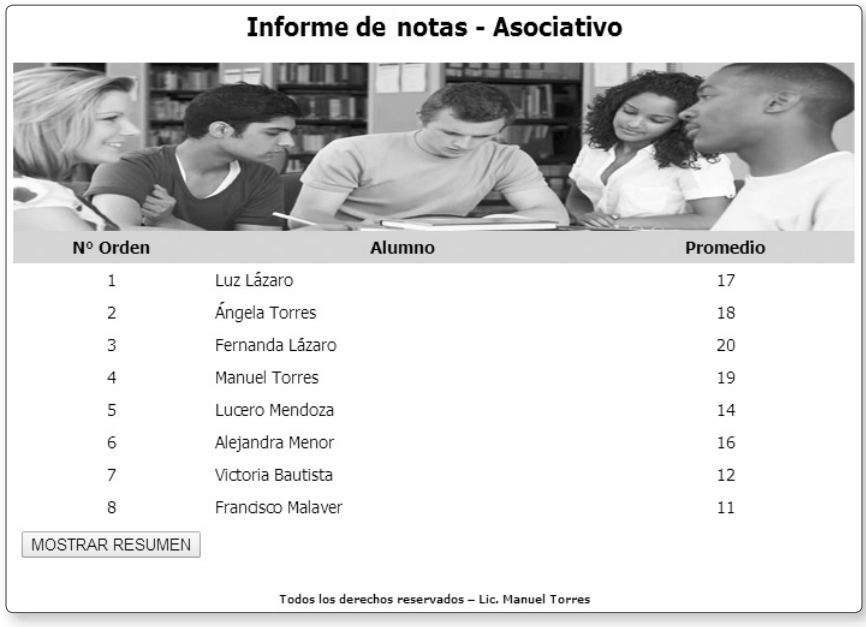
Finalmente, llegamos a la impresión de los nombres de aquellos alumnos que obtuvieron el mayor y menor promedio; esto se realiza mediante una consulta directa al arreglo y su índice, de tal modo que **getAlumnos()[$maIndice]** devuelve el nombre del alumno según el índice mayor encontrado en la función **índice_maximo_minimo**. De la misma forma es aplicado al nombre del alumno con menor promedio.

- **Caso desarrollado 2: Arreglo asociativo – Informe de notas**

Implemente una aplicación web con PHP que permita generar informe de notas según la siguiente tabla:

NOMBRES Y APELLIDOS	PROMEDIO
Luz Lázaro	17
Ángela Torres	18
Fernanda Lázaro	20
Manuel Torres	19
Lucero Mendoza	14
Alejandra Menor	16
Victoria Bautista	12
Francisco Malaver	11

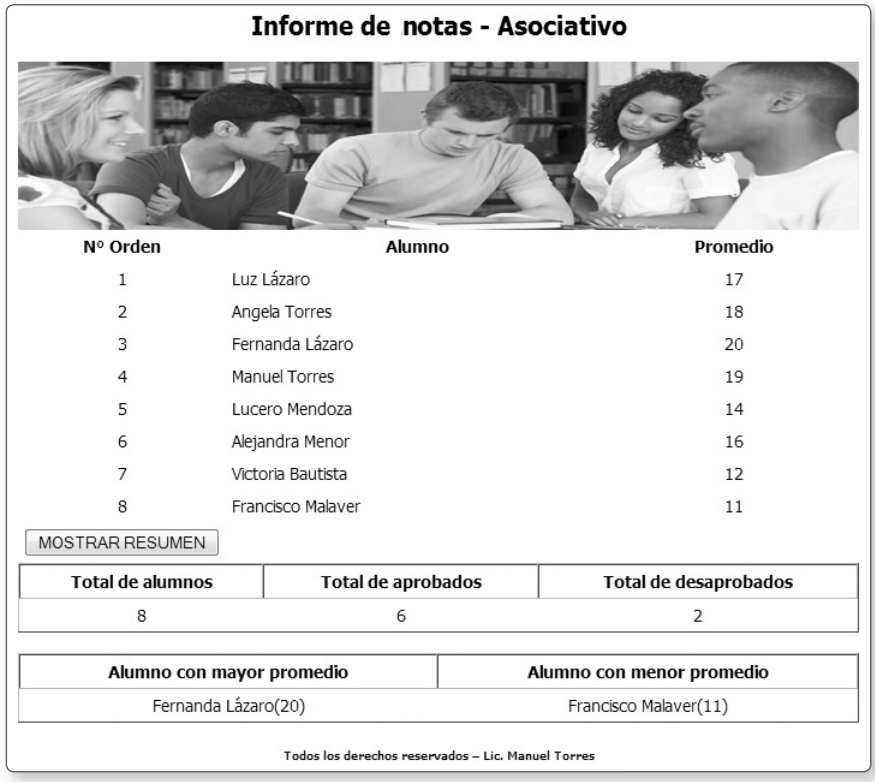
La interfaz gráfica inicial es la siguiente:



Sólo fines educativos - FreeLibros

Tener en cuenta:

- Al presionar el botón **Mostrar resumen** se mostrará el **Total de alumnos registrados**, **Total de alumnos aprobados**, **Total de alumnos desaprobados**, nombre del alumno con mayor promedio y nombre del alumno con menor promedio.
- Use funciones implementadas por el usuario para todos los procesos de cálculo.
- La presentación final de la aplicación es de la siguiente manera:



Archivo: **notas_a.php**

```
<!DOCTYPE html>
<html>
    <head>
        <meta charset="UTF-8">
        <title>Informe de Notas</title>
        <link href="estilo.css" rel="stylesheet">
    </head>
    <body>
        <header>
            <h2 id="centrado">Informe de Notas - Asociativo</h2>
            <img src="bannerEstudiantes.jpg" width="700" height="140"/>
        </header>
        <section>
          <form action="notas_a.php" method="POST">
            <table border="0" width="700" cellspacing="0" cellpadding="5">
                <tr>
                    <th width="10">N° Orden</th>
                    <th>Alumno</th>
                    <th>Promedio</th>
```

Sólo fines educativos - FreeLibros

```
            </tr>
            <?php
                //Imprimir el listado de alumnos
                $i=1;
                foreach(getAlumnos() as $alumno => $promedio){
            ?>
            <tr>
                <td id="centrado"><?php echo $i; ?></td>
                <td><?php echo $alumno; ?></td>
                <td id="centrado"><?php echo $promedio; ?></td>
            </tr>
                <?php
                    $i++;
                }
                ?>
            <tr>
                <td><input type="submit" value="MOSTRAR RESUMEN"
                           name="btnMostrar"/>
                </td>
                <td></td>
                <td></td>
            </tr>
        </table>
    </form>
        <?php
          //Determinar el total de aprobados y desaprobados
          list($tAprobados,$tDesaprobados)=total_aprobados_desaprobados();

          //Condicionar la muestra de resultados
          if(isset($_POST["btnMostrar"])){
        ?>
        <table border="1" width="700" cellspacing="0" cellpadding="5">
            <tr>
                <th>Total de alumnos</th>
                <th>Total de aprobados</th>
                <th>Total de desaprobados</th>
            </tr>
            <tr>
                <td id="centrado"><?php echo getTotal(); ?></td>
                <td id="centrado"><?php echo $tAprobados; ?></td>
                <td id="centrado"><?php echo $tDesaprobados; ?></td>
            </tr>
        </table>
        <?php
            //Determinar el mas alto promedio con el nombre del alumno
            list($mayorAlumno,$maximo) = determinaAltoPromedio();
            list($menorAlumno,$minimo) = determinaBajoPromedio();
        ?>
        <br />
        <table border="1" width="700" cellspacing="0" cellpadding="5">
            <tr>
                <th>Alumno con mayor promedio</th>
                <th>Alumno con menor promedio</th>
            </tr>
            <tr>
                <td id="centrado">
                       <?php echo $mayorAlumno.'('.$maximo.')'; ?>
                </td>
                <td id="centrado">
                       <?php echo $menorAlumno.'('.$minimo.')'; ?>
                </td>
            </tr>
```

Sólo fines educativos - FreeLibros

```
            </table>
                <?php } ?>
        </section>
        <footer>
            <h6 id="centrado">Todos los derechos reservados –
                Lic. Manuel Torres</h6>
        </footer>
    </body>
</html>
<?php
    //Función de implementación para el arreglo asociativo de alumnos
    function getAlumnos(){
        return array('Luz Lázaro'=>17,'Angela Torres'=>18,
                     'Fernanda Lázaro'=>20,'Manuel Torres'=>19,
                     'Lucero Mendoza'=>14,'Alejandra Menor'=>16,
                     'Victoria Bautista'=>12,'Francisco Malaver'=>11);
    }

    //Función que determina el total de alumnos
    function getTotal(){
        return count(getAlumnos());
    }

    //Función que determina el total de aprobados y desaprobados
    function total_aprobados_desaprobados(){
        $tAprobados = 0;
        $tDesaprobados = 0;
        foreach (getAlumnos() as $alumno => $promedio) {
            if ($promedio<13)
                $tDesaprobados++;
                else
                $tAprobados++;
        }
        return array($tAprobados,$tDesaprobados);
    }

    //Determinar el mas alto promedio y a que alumno pertenece
    function determinaAltoPromedio(){
        $alumnos = getAlumnos();
        arsort($alumnos);
        reset($alumnos);
        list($alumno,$promedio)=each($alumnos);
        return array($alumno,$promedio);
    }
    //Determinar el mas bajo promedio y a que alumno pertenece
    function determinaBajoPromedio(){
        $alumnos = getAlumnos();
        asort($alumnos);
        reset($alumnos);
        list($alumno,$promedio)=each($alumnos);
        return array($alumno,$promedio);
    }
?>
```

Sólo fines educativos - FreeLibros

Comentarios:

```
function getAlumnos(){
   return array('Luz Lázaro'=>17,'Angela Torres'=>18,
                'Fernanda Lázaro'=>20,'Manuel Torres'=>19,
                'Lucero Mendoza'=>14,'Alejandra Menor'=>16,
                'Victoria Bautista'=>12,'Francisco Malaver'=>11);
}
```

getAlumnos permite implementar y devolver un arreglo de alumnos y promedios de tipo asociativo, a diferencia del arreglo indexado, que se tenía que realizar en dos arreglos llamados **alumnos** y **promedios**. El asociativo lo realiza en una sola, donde la clave es el nombre del alumno, mientras que el valor es el promedio de cada alumno.

```
function getTotal(){
      return count(getAlumnos());
}
```

getTotal determina el total de alumnos registrados en el arreglo mediante la función **count**.

```
function total_aprobados_desaprobados(){
     $tAprobados = 0;
     $tDesaprobados = 0;
     foreach (getAlumnos() as $alumno => $promedio) {
        if ($promedio<13)
            $tDesaprobados++;
            else
            $tAprobados++;
     }
     return array($tAprobados,$tDesaprobados);
}
```

total_aprobados_desaprobados permite determinar el total de alumnos aprobados y desaprobados desde el arreglo asociativo. Para poder recorrer el arreglo se tiene que implementar la función **foreach**, asignando a la variable **$alumno** el nombre del alumno; y a **$promedio**, el promedio de notas de cada alumno. A partir de aquí es donde se determina si el alumno se encuentra aprobado o no. Finalmente la función devuelve un arreglo con los contadores de aprobados y desaprobados.

```
function determinaAltoPromedio(){
     $alumnos = getAlumnos();
     arsort($alumnos);
     reset($alumnos);
     list($alumno,$promedio)=each($alumnos);
     return array($alumno,$promedio);
}
```

La función **determinaAltoPromedio** permite devolver el nombre del alumno y el promedio más alto encontrado en el arreglo; hay que tener en cuenta que esta función primero ordena en forma descendente los elementos (**arsort**) y luego captura el primer elemento encontrado. Para esto se ubica en el punto inicial con (**reset**), luego la función **list** registra el nombre del alumno y el promedio encontrado.

Sólo fines educativos - FreeLibros

```
function determinaBajoPromedio(){
      $alumnos = getAlumnos();
      asort($alumnos);
      reset($alumnos);
      list($alumno,$promedio)=each($alumnos);
      return array($alumno,$promedio);
}
```

La función **determinaBajoPromedio** permite devolver el nombre del alumno con promedio más bajo; de igual manera que el proceso anterior, ordenaremos de forma ascendente (**asort**), luego nos ubicamos en el inicio del arreglo (**reset**) y, finalmente, obtenemos la información mediante la función **list** para registrarlos en las variables **$alumno** y **$promedio**.

```
<?php
  $i=1;
  foreach(getAlumnos() as $alumno => $promedio){
?>
<tr>
   <td id="centrado"><?php echo $i; ?></td>
   <td><?php echo $alumno; ?></td>
   <td id="centrado"><?php echo $promedio; ?></td>
</tr>
<?php
   $i++;
   }
?>
```

El script permite recorrer por todo el arreglo, usando la estructura **for each**; desde aquí se podrá imprimir los valores del arreglo en sus celdas correspondientes.

```
<?php
      list($tAprobados,$tDesaprobados)=total_aprobados_desaprobados();
      if(isset($_POST["btnMostrar"])){
?>
<table border="1" width="700" cellspacing="0" cellpadding="5">
```

Para obtener el total de aprobados y desaprobados invocamos a la función **total_aprobados_desaprobados**, que devuelve una matriz de dos elementos y lo recibimos en las variables **$tAprobados** y **$tDesaprobados** respectivamente.

```
<tr>
    <td id="centrado"><?php echo getTotal(); ?></td>
    <td id="centrado"><?php echo $tAprobados; ?></td>
    <td id="centrado"><?php echo $tDesaprobados; ?></td>
</tr>
```

En el script imprimiremos todos los valores obtenidos en sus celdas correspondientes, como el **getTotal**, que devuelve el total de alumnos registrados; **tAprobados**, que devuelve el total de aprobados; y **tDesaprobados**, que devuelve el total de desaprobados registrados en el arreglo.

Sólo fines educativos - FreeLibros

```
<?php
    list($mayorAlumno,$maximo) = determinaAltoPromedio();
    list($menorAlumno,$minimo) = determinaBajoPromedio();
?>
```

Asignaremos a las variables **$mayorAlumno** y **$maximo** los valores obtenidos desde la función **determinaAltoPromedio**; hay que tener en cuenta que esta función devuelve una matriz. De la misma manera se asignará a las variables **$menorAlumno** y **$minimo** desde la función **determinaBajoPromedio**.

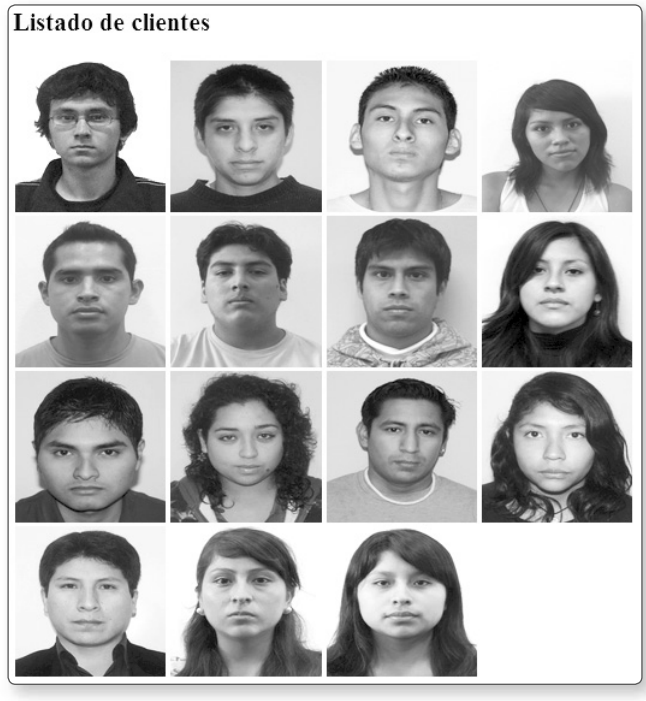
```
<tr>
    <td id="centrado">
        <?php echo $mayorAlumno.'('.$maximo.')'; ?>
    </td>
    <td id="centrado">
        <?php echo $menorAlumno.'('.$minimo.')'; ?>
    </td>
</tr>
```

Imprimiremos las variables **$mayorAlumno** y **$menorPromedio** en sus respectivas celdas concatenado el nombre y el promedio del alumno.

- **Caso desarrollado 3: Arreglo indexado – Manejo de imágenes**

  Implemente una aplicación web con PHP que permita listar las imágenes de los clientes en cuatro columnas con un tamaño de imagen 140x140px.

  La interfaz gráfica propuesta es la siguiente:



Sólo fines educativos - FreeLibros

Debemos considerar los siguientes aspectos:

- Antes de empezar con la aplicación debe contar con imágenes de los clientes, colocarlas en una carpeta, para nuestro caso le hemos asignado el nombre CLIENTES, tal como se muestra en la siguiente imagen:

- Dicha carpeta CLIENTES debe encontrarse en la carpeta del servidor, para nuestro caso la ruta es la siguiente: WAMP>WWW>Capitulo7>.
- Los nombres de los archivos deben ser códigos, los cuales serán las claves para el acceso de la imagen, para nuestro caso se le asignó c0001 en forma consecutiva.

Archivo: **imagenBasica.php**

```
<!DOCTYPE html>
<html>
    <head>
        <meta charset="UTF-8">
        <title>Listado de Clientes</title>
    </head>
    <body>
        <header>
            <h2 id="centrado">Listado de clientes</h2>
        </header>
        <section>
            <?php
                $clientes=  array('c0001','c0002','c0003',
                                  'c0004','c0005','c0006',
                                  'c0007','c0008','c0009',
                                  'c0010','c0011','c0012',
                                  'c0013','c0014','c0015')
            ?>
            <table border="0" width="700" cellspacing="1" cellpadding="1">
                <tr>
                    <td>
                    <?php
                        $i=1;
                        foreach ($clientes as $imagen) {
                     ?>
                        <img src="clientes/<?php echo $imagen;?>.jpg"
                             width="140" height="140"/>
                    <?php
                        if ($i%4==0) echo '<br>';
                        $i++;
```

Sólo fines educativos - FreeLibros

```
                        }
                    ?>
                    </td>
                </tr>
            </table>
        </section>
        <footer>
            <h6 id="centrado">Todos los derechos reservados –
                Lic. Manuel Torres</h6>
        </footer>
    </body>
</html>
```

Comentarios:

```
$clientes=  array('c0001','c0002','c0003',
                  'c0004','c0005','c0006',
                  'c0007','c0008','c0009',
                  'c0010','c0011','c0012',
                  'c0013','c0014','c0015')
```

La variable de tipo arreglo **$clientes** almacena los códigos consecutivos desde c0001 hasta c0015; estos representan al nombre de cada uno de los archivos de tipo imagen de los clientes.

```
<?php
       $i=1;
       foreach ($clientes as $imagen) {
?>
<img src="clientes/<?php echo $imagen;?>.jpg"  width="140" height="140"/>
<?php
       if ($i%4==0)
           echo '<br>';
           $i++;
       }
?>
```

Mediante la estructura **foreach** accederemos a todos los elementos del arreglo, y así podremos imprimir las imágenes desde la carpeta **clientes** con el siguiente código img src="clientes/<?php echo $imagen;?>.jpg".

Para imprimir de cuatro en cuatro las imágenes, usamos la variable **$i**, al cual se le condiciona que sea divisible por cuatro; si es así, se cambiará de línea con la función <br>.

Sólo fines educativos - FreeLibros

- **Caso desarrollado 4: Arreglo indexado – Paginación de productos**

Implemente una aplicación web con PHP que permita listar los productos registrados en un arreglo indexado, de tal manera que muestre cada 5 productos.

La interfaz gráfica de la primera página tiene el siguiente formato:

LISTADO DE PRODUCTOS

DESCRIPCIÓN DEL PRODUCTO	PRECIO DEL PRODUCTO
Lavadora	S/. 1500.00
Radiograbadora	S/. 500.00
Licuadora	S/. 400.00
Extractora	S/. 700.00
Afeitadora	S/. 80.00
1 2 3 4 Página siguiente	

Todos los derechos reservados – Lic. Manuel Torres

La interfaz gráfica de la segunda página tiene el siguiente formato:

LISTADO DE PRODUCTOS

DESCRIPCIÓN DEL PRODUCTO	PRECIO DEL PRODUCTO
Lampara	S/. 50.00
Cocina	S/. 1300.00
Lavavajillas	S/. 170.00
Batidora	S/. 100.00
Secadora	S/. 1000.00
Página anterior 1 **2** 3 4 Página Siguiente	

Todos los derechos reservados – Lic. Manuel Torres

La interfaz gráfica de la tercera página tiene el siguiente formato:

LISTADO DE PRODUCTOS

DESCRIPCIÓN DEL PRODUCTO	PRECIO DEL PRODUCTO
Tostadora	S/. 60.00
Aspiradora	S/. 250.00
Televisor	S/. 2500.00
Campana	S/. 700.00
Microondas	S/. 800.00
Página anterior 1 2 **3** 4 Página siguiente	

Todos los derechos reservados – Lic. Manuel Torres

Sólo fines educativos - FreeLibros

La interfaz gráfica de la cuarta página tiene el siguiente formato:

LISTADO DE PRODUCTOS

DESCRIPCIÓN DEL PRODUCTO	PRECIO DEL PRODUCTO
Plancha	S/. 150.00
Calentador	S/. 1200.00
Cafetera	S/. 50.00
Página anterior 1 2 3 **4**	

Todos los derechos reservados – Lic. Manuel Torres

Archivo: **paginacion.php**

```
<!DOCTYPE html>
<html>
    <head>
        <meta charset="UTF-8">
        <title>Paginación de Productos</title>
        <link href="estilo.css" rel="stylesheet">
    </head>
    <body>
        <header>
            <h2 id="centrado">LISTADO DE PRODUCTOS</h2>
        </header>
        <section>
        <?php
            error_reporting(0);
            //Arreglo de productos
            $productos = array( 'Lavadora'=>1500,'Radiograbadora'=>500,
                                'Licuadora'=>400,'Extractora'=>700,
                                'Afeitadora'=>80,'Lampara'=>50,
                                'Cocina'=>1300,'Lavavajillas'=>170,
                               'Batidora'=>100,'Secadora'=>1000,
                               'Tostadora'=>60,'Aspiradora'=>250,
                               'Televisor'=>2500,'Campana'=>700,
                               'Microondas'=>800,'Plancha'=>150,
                               'Calentador'=>1200,'Cafetera'=>50);

            //Determinar si hay páginas que mostrar
            if (isset($_GET[pagina]))
                $pagina = $_GET[pagina];
                else
                $pagina=1;

            //Invocar a la función de paginación
            paginar($productos, 5, $pagina);
         ?>
        </section>
        <footer>
            <h6 id="centrado">Todos los derechos reservados –
                Lic. Manuel Torres</h6>
        </footer>
    </body>
</html>
<?php
    //Implementación de la función paginación
    function paginar($misProductos, $total, $pagina) {
```

Sólo fines educativos - FreeLibros

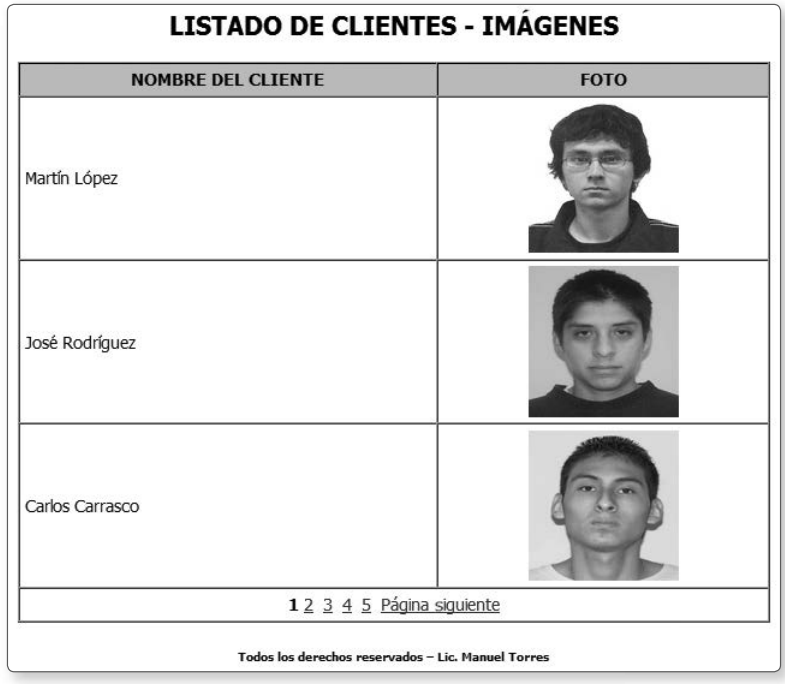
```
        //Determinar el total de paginas que genera
        //la cantidad total de productos
        $paginas = ceil(count($misProductos) / $total);

        //Configurando el inicio de la paginación
        $inicio = ($pagina-1)*$total;

        //Configurando la finalización de la paginación
        $final = $pagina*$total;

        //Listando los productos en una tabla
    echo '<table border="1" width="700" cellspacing="0" cellpadding="5">';
        echo '<tr>';

        //Obtener los N registros
        $paginado=array_slice($misProductos,$inicio,$final);

        echo "<tr><td>DESCRIPCION DEL PRODUCTO</td>";
        echo "<td>PRECIO DEL PRODUCTO</td></tr>";

        //Listando los productos y sus precios
        for ($i=$inicio; $i<$final; $i++) {
            list($producto,$precio) = each($paginado);
            if (isset($misProductos[$producto])){
                echo "<tr><td>$producto</td>";
                echo "<td>S/. ".number_format($precio,'2','.','');".
                      </td></tr>";
            }
            else
                break;
        }
        echo '</tr>';

        //Mostrando las paginas
        echo '<tr><td colspan="2" id="centrado">';
        if ($pagina>1)
            echo "<a href=\"paginacion.php?pagina=".($pagina-1)."\">
                     Página Anterior</a> ";
            for ($i=1; $i<=$paginas; $i++) {
                if ($i == $pagina)
                    echo "<strong>$i</strong> ";
                    else
                    echo "<a href=\"paginacion.php?pagina=$i\">$i
                          </a>  ";
            }

            if ($pagina<$paginas)
                echo "<a href=\"paginacion.php?pagina=" . ($pagina+1) .
"\">
                            Página Siguiente</a>";
        echo '</td></tr>';
        echo '</table>';
        return;
    }
?>
```

Sólo fines educativos - FreeLibros

- **Caso desarrollado 5: Arreglo indexado – Paginación de imágenes**

  Implemente una aplicación web con PHP que permita listar las imágenes de los clientes de tal forma que solo se muestren 3 clientes por página, dicha imagen cuenta con un tamaño de 140 x 140px.

  La interfaz gráfica de la primera página tiene el siguiente formato:



La interfaz gráfica de la segunda página tiene el siguiente formato:

Sólo fines educativos - FreeLibros

La interfaz gráfica de la tercera página tiene el siguiente formato:

La interfaz gráfica de la cuarta página tiene el siguiente formato:

Sólo fines educativos - FreeLibros

La interfaz gráfica de la quinta página tiene el siguiente formato:

Archivo: **paginacion_imagenes.php**

```
<!DOCTYPE html>
<html>
    <head>
        <meta charset="UTF-8">
        <title>Paginación de Clientes con Imágenes</title>
        <link href="estilo.css" rel="stylesheet">
    </head>
    <body>
        <header>
            <h2 id="centrado">LISTADO DE CLIENTES - IMAGENES</h2>
        </header>
        <section>
        <?php
            error_reporting(0);
            //Arreglo de clientes
            $clientes=  array('c0001'=>'Martin López',
                              'c0002'=>'José Rodríguez',
                              'c0003'=>'Carlos Carrasco',
                              'c0004'=>'Karla Rojas',
                              'c0005'=>'Manuel Fernández',
                              'c0006'=>'Ricardo Ríos',
                              'c0007'=>'Fernando Gutiérrez',
                              'c0008'=>'Lizbeth García',
                              'c0009'=>'Guillermo Gómez',
                              'c0010'=>'María Hilario',
                              'c0011'=>'Maco Díaz',
                              'c0012'=>'Karina Gálvez',
                              'c0013'=>'Ricardo Rivera',
```

Sólo fines educativos - FreeLibros

```
                                'c0014'=>'Julia Rubio',
                                'c0015'=>'María Celedonio');

            //Determinar si hay páginas que mostrar
            if (isset($_GET[pagina]))
                $pagina = $_GET[pagina];
                else
                $pagina=1;

            //Invocar a la función de paginación
            paginar($clientes, 3, $pagina);
         ?>
        </section>
        <footer>
            <h6 id="centrado">Todos los derechos reservados –
                Lic. Manuel Torres</h6>
        </footer>
    </body>
</html>
<?php
    //Implementación de la función paginación
    function paginar($misClientes, $total, $pagina) {
        //Determinar el total de paginas que genera
        //la cantidad total de productos
        $paginas = ceil(count($misClientes) / $total);

        //Configurando el inicio de la paginación
        $inicio = ($pagina-1)*$total;

        //Configurando la finalización de la paginación
        $final = $pagina*$total;

        //Listando los productos en una tabla
        echo '<table border="1" width="700" cellspacing="0" cellpadding="5">';
        echo '<tr>';

        //Obtener los N registros
        $paginado=array_slice($misClientes,$inicio,$final);

        echo "<tr><th>NOMBRE DEL CLIENTE</th>";
        echo "<th>FOTO</th></tr>";

        //Listando los clientes y sus imágenes
        for ($i=$inicio; $i<$final; $i++) {
            list($codigo,$nombre) = each($paginado);
            if (isset($misClientes[$código])){
                echo "<tr><td>$nombre</td>";
                echo '<td><img src="clientes/'.$codigo.'.jpg"
                                width="140" height="140"/></td></tr>';
            }
            else
                break;
        }
        echo '</tr>';

        //Mostrando las paginas
        echo '<tr><td colspan="2" id="centrado">';
        if ($pagina>1)
```

Sólo fines educativos - FreeLibros

```
            echo "<a href=\"paginacion_imagenes.php?pagina=".($pagina-1)."\">
                     Página Anterior</a> ";
            for ($i=1; $i<=$paginas; $i++) {
                if ($i == $pagina)
                    echo "<strong>$i</strong> ";
                    else
                    echo "<a href=\"paginacion_imagenes.php?pagina=$i\">$i
                          </a>  ";
            }

            if ($pagina<$paginas)
               echo "<a href=\"paginacion_imagenes.php?pagina=".
                     ($pagina+1). "\">Página Siguiente</a>";
        echo '</td></tr>';
        echo '</table>';
        return;
    }
?>
```

- **Caso desarrollado 6: Manejo de la función include – Listado de productos**

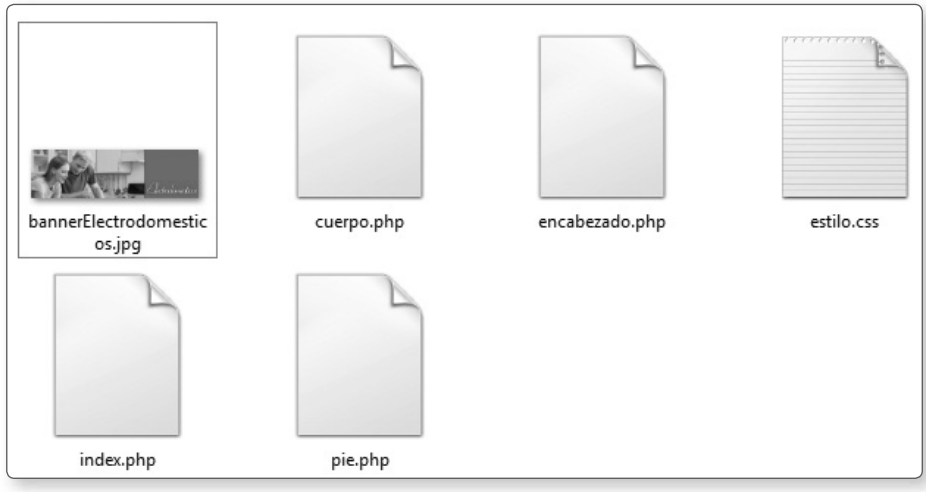
  Implemente una aplicación web con PHP que permita listar los productos y sus precios de acuerdo a la siguiente tabla:

PRODUCTOS	PRECIO
Lavadora	$1500.00
Radiograbadora	$500.00
Licuadora	$400.00
Extractora	$700.00
Afeitadora	$80.00
Lámpara	$50.00
Cocina	$1300.00
Lavavajillas	$170.00
Batidora	$100.00
Secadora	$1000.00
Tostadora	$60.00
Aspiradora	$250.00
Televisor	$2500.00
Campana	$700.00
Microondas	$800.00
Plancha	$150.00
Calentador	$1200.00
Cafetera	$50.00

Sólo fines educativos - FreeLibros

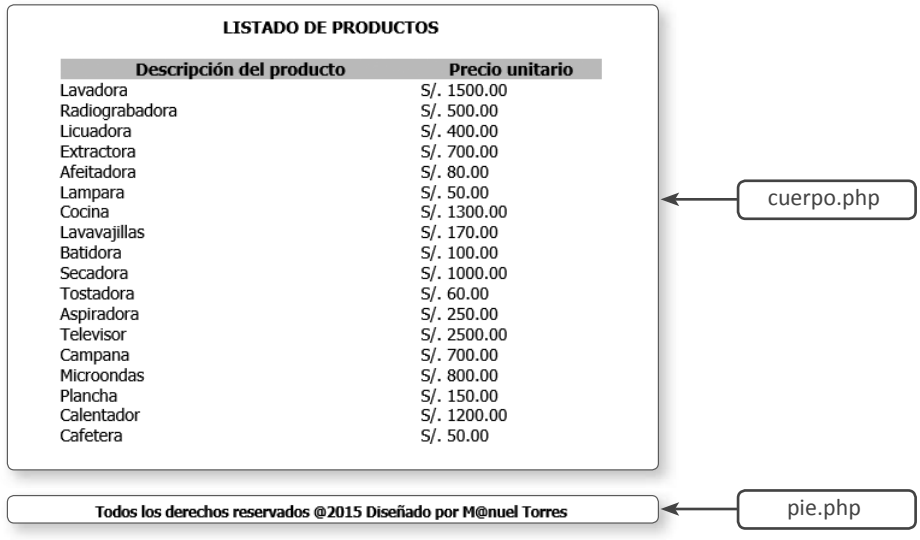
Tener en cuenta:

- Los productos y los precios se almacenan en un arreglo asociativo.
- Crear una carpeta llamada **Tienda** que contenga las imágenes y archivos php del proyecto, al final deberá mostrarse de la siguiente manera:



- El archivo **index.php** tiene la composición del documento web.
- El archivo **estilo.css** propone los formatos, colores y estilos del documento web.
- El archivo **cuerpo.php** muestra la información de los productos y sus precios.
- El archivo **ecabezado.php** muestra la información correspondiente al escabezado de la página principal; aquí se mostrará el banner del documento.
- El archivo **pie.php** muestra la información correspondiente al pie de página, el cual muestra información referente al autor.
- Debemos tener la siguiente distribución de documentos:

Sólo fines educativos - FreeLibros



La interfaz gráfica propuesta es la siguiente:

Sólo fines educativos - FreeLibros

Archivo: **encabezado.php**

```
<!DOCTYPE html>
<html>
    <head>
        <meta charset="UTF-8">
        <title></title>
    </head>
    <body>
        <img src="bannerElectrodomesticos.jpg"
             width="700" height="200"/>
    </body>
</html>
```

Archivo: **pie.php**

```
<!DOCTYPE html>
<html>
    <head>
        <meta charset="UTF-8">
        <title></title>
    </head>
    <body>
        <h5 id="centrado">Todos los derechos reservados @2015
                          Diseñado por M@nuel Torres</h5>
    </body>
</html>
```

Archivo: **cuerpo.php**

```
<!DOCTYPE html>
<html>
    <head>
        <meta charset="UTF-8">
        <title></title>
    </head>
    <body>
        <?php
            //Arreglo de productos
            $productos = array( 'Lavadora'=>1500,'Radiograbadora'=>500,
                                'Licuadora'=>400,'Extractora'=>700,
                                'Afeitadora'=>80,'Lampara'=>50,
                                'Cocina'=>1300,'Lavavajillas'=>170,
                                'Batidora'=>100,'Secadora'=>1000,
                                'Tostadora'=>60,'Aspiradora'=>250,
                                'Televisor'=>2500,'Campana'=>700,
                                'Microondas'=>800,'Plancha'=>150,
                                'Calentador'=>1200,'Cafetera'=>50);

            //Imprimir el listado de productos
            echo '<table border="0" cellspacing="0" cellpadding="0">';
            echo '<tr>';
            echo '<th width=300>Descripción del producto</th>';
            echo '<th width=150>Precio unitario</th>';
            echo '</tr>';

            for($i=0;$i<count($productos);$i++){
                list($descripcion,$precio)=each($productos);
                echo '<tr>';
                echo "<td>$descripcion</td>";
                echo '<td>S/. '.number_format($precio,'2','.','').'</td>';
                echo '</tr>';
```

Sólo fines educativos - FreeLibros

```
            }
            echo '</table>';
        ?>
    </body>
</html>
```

Archivo: **estilo.css**

```
body{
    font-family: tahoma;
    font-size: 14px;
}
#centrado{
    text-align: center;
}
table {
    margin: auto;
}
img{
    margin: auto;
    display: block;
}
table th {
background-color: #b9c9fe;
}
```

Archivo: **index.php**

```
<!DOCTYPE html>
<html>
    <head>
        <meta charset="UTF-8">
        <title>CONTROL DE VENTAS</title>
        <link href="estilo.css" rel="stylesheet">
    </head>
    <body>
        <header>
            <?php include('encabezado.php'); ?>
        </header>
        <section>
            <h4 id="centrado">LISTADO DE PRODUCTOS</h4>
            <?php include('cuerpo.php');?>
        </section>
        <footer>
            <?php include('pie.php'); ?>
        </footer>
    </body>
</html>
```

- **Caso desarrollado 7: Manejo de la función require – Control de pago**

  Implemente una aplicación web con PHP que permita controlar el pago de los vendedores de tal manera que se deba registrar el nombre del vendedor y el monto vendido, para luego mostrar el monto por comisión, monto de descuento y monto neto a un determinado vendedor.

Sólo fines educativos - FreeLibros

La interfaz gráfica propuesta es la siguiente:

Tener en cuenta:

- Los valores registrados por el usuario deben mantenerse luego de presionar el botón **Procesar**.
- El porcentaje de comisión se basa en el monto vendido de acuerdo a la siguiente cuadro:

MONTO VENDIDO	PORCENTAJE DE COMISIÓN
≤ 5000	10 %
> 5000 y ≤ 10 000	20 %
> 10 000 y ≤ 15 000	30 %
> 15 000	50 %

Archivo: **index.php**

```
<!DOCTYPE html>
<html>
    <head>
        <meta charset="UTF-8">
        <title>Control de Pago - Vendedores</title>
        <link href="estilo.css" rel="stylesheet">
    </head>
    <body>
        <header>
            <h2 id="centrado">
                CONTROL DE PAGO - VENDEDORES
            </h2>
            <img src="banner_employees.jpg"
                 width="700" height="200" />
        </header>
        <section>
            <?php
                require 'capturaDatos.php';
                require 'calculos.php';
            ?>
            <form name="frmPago" method="POST">
                <table border="0" width="700" cellspacing="0" cellpadding="5">
                    <tr>
                        <td>Vendedor</td>
```

Sólo fines educativos - FreeLibros

```
                        <td><input type="text" name="txtVendedor" size="70"
                                value="<?php echo getVendedor(); ?>" />
                        </td>
                    </tr>
                    <tr>
                        <td>Monto vendido</td>
                        <td><input type="text" name="txtMonto" size="30"
                                value="<?php echo getMonto(); ?>" />
                        </td>
                    </tr>
                    <tr>
                        <td><input type="submit" value="PROCESAR" /></td>
                        <td></td>
                    </tr>
                    <tr>
                      <td>Monto por comisión</td>
                      <td>
                       <?php $montoComision=determinaComision(getMonto());
                         echo 'S/.'.number_format($montoComision,'2','.','');
                       ?>
                      </td>
                    </tr>
                    <tr>
                     <td>Monto de descuento</td>
                     <td>
                       <?php
                         $montoDescuento=determinaDescuento($montoComision);
                         echo 'S/.'.number_format($montoDescuento,'2','.','');
                       ?>
                      </td>
                    </tr>
                    <tr>
                     <td>Monto neto</td>
                     <td>
                       <?php
                  $montoNeto=calculaMontoNeto($montoComision,$montoDescuento);
                           echo 'S/. '.number_format($montoNeto,'2','.','');
                       ?>
                     </td>
                    </tr>
                </table>

            </form>
        </section>
        <footer>
            <h5 id="centrado">Todos los derechos reservados @2015
                        Diseñado por M@nuel Torres</h5>
        </footer>
    </body>
</html>
```

Archivo: **calculos.php**

```
<?php
    //Implementacion de funciones para los calculos en el proceso
    function determinaComision($monto){
        if ($monto<=5000)
            return $monto*0.10;
        elseif ($monto<=10000)
            return $monto*0.20;
        elseif ($monto<=15000)
            return $monto*0.30;
        else
            return $monto*0.5;
```

Sólo fines educativos - FreeLibros

```
    }

    function determinaDescuento($comision){
        if($comision<1800)
            return 0;
        else
            return $comision*0.12;
    }

    function calculaMontoNeto($comision,$descuento){
        return $comision-$descuento;
    }
?>
```

Archivo: **capturaDatos.php**

```
<?php
    //Implementacion de funciones que capturan valor
    function getVendedor(){
        return $_POST['txtVendedor'];
    }
    function getMonto(){
        return $_POST['txtMonto'];
    }
?>
```

- **Caso desarrollado 8: Manejo de la función require – Control de facturas**

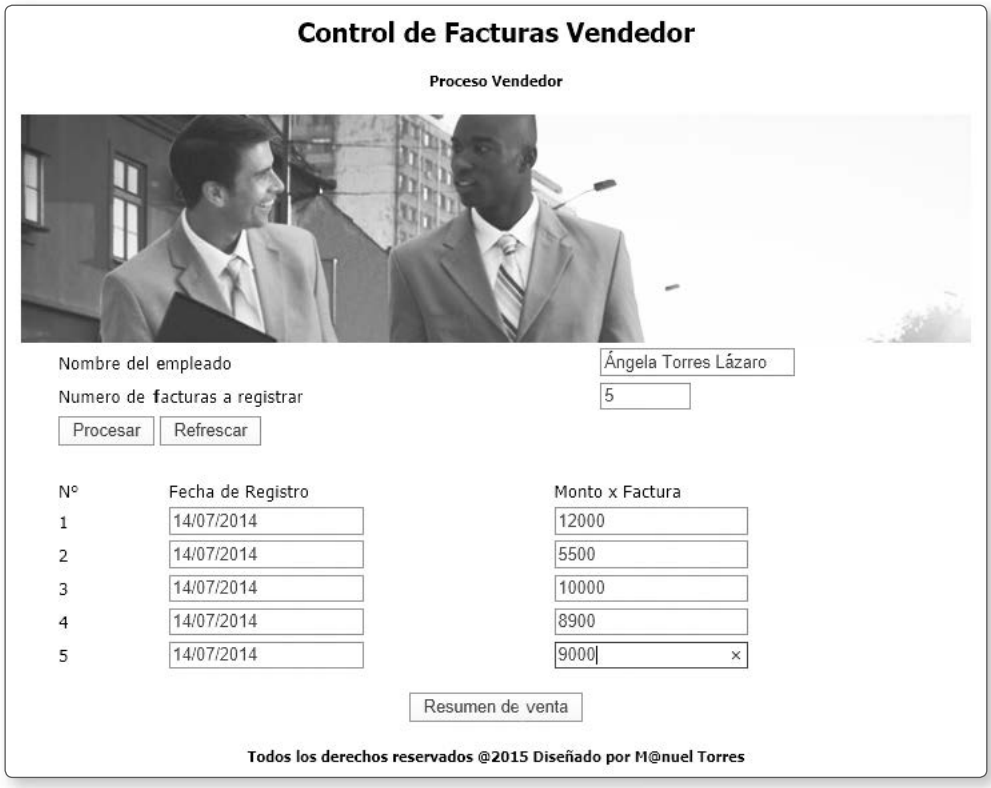
Implemente una aplicación web con PHP que permita controlar las facturas que tiene que registrar un determinado empleado, quien ingresará su nombre completo y el número de facturas a registrar.

La interfaz gráfica propuesta es la siguiente:

Archivo: **index.php**

Sólo fines educativos - FreeLibros

Al iniciar la aplicación solo se mostrarán las casillas de ingreso del nombre del empleado y el número de facturas a registrar; solo al presionar el botón **Resumen de venta** se mostrarán las casillas que permitirán el ingreso de los montos por factura con la fecha actual y el número correlativo de registro, tal como se muestra en la siguiente imagen:



Una vez registrados los montos por facturas, se deberá generar un resumen de venta mediante un botón, el cual listará el nombre del empleado, el total de la venta y el monto de comisión que asciende a un 15 % aplicado al total de venta, tal como se muestra en la siguiente imagen:

Archivo: **resumen.php**

Sólo fines educativos - FreeLibros

Archivo: **index.php**

```
<!DOCTYPE html>
<html>
    <head>
        <meta charset="UTF-8">
        <title></title>
        <link href="estilo.css" rel="stylesheet">
    </head>
    <body>
        <header>
       <h2 id="centrado">Control de Facturas Vendedor</h2>
       <h5 id="centrado">Proceso Vendedor</h5>
          <img src="bannerVendedor.jpg" width="750" height="180"/>
        </header>
        <section>
            <?php
                require 'capturaDatos.php';
            ?>
            <form name="frmFacturas" method="POST" action="index.php">
              <table width="700" border="0" cellspacing="5" cellpadding="0">
                    <tr>
                        <td width="264">Nombre del Empleado</td>
                        <td width="164">
                            <input type="text" name="txtVendedor"
                                   value="<?php echo getVendedor();?>">
                        </td>
                    </tr>
                    <tr>
                        <td>Numero de Facturas a registrar</td>
                        <td align="left">
                            <input name="txtN" type="text"
                                  size="6" value="<?php echo getNumero();?>">
                        </td>
                    </tr>
                    <tr>
                        <td colspan="2">
                          <input type="submit" name="btnProcesar"
                                 value="Procesar">
                          <input type="reset" name="btnRefrescar"
                                 value="Refrescar">
                        </td>
                    </tr>
                </table>
                <br />
            </form>

            <form name="frmResumen" method="POST" action="resumen.php">
                <table  width="700" height="41" border="0"
                        cellpadding="0" cellspacing="5">
                    <tr>
                     <td width="42">N°</td>
                     <td width="112">Fecha de Registro</td>
                     <td width="132">Monto x Factura</td>
                    </tr>
                 <?php
                      //Mostrar las cajas de texto dinámicas
                      for($i=1;$i<=getNumero();$i++){
                 ?>
                    <tr>
                        <td><?php echo $i; ?></td>
                        <td>
```

Sólo fines educativos - FreeLibros

```
                            <input name="txtFecha" type="text"
                                   value="<?php echo date("d/m/Y");?>">
                        </td>
                        <td><input name="txtMonto[]" type="text"></td>
                    </tr>
                  <?php } ?>
                </table>

                <input type="hidden" name="txtNumeroOculto"
                        value="<?php echo getNumero();?>">
                <input type="hidden" name="txtVendedorOculto"
                        value="<?php echo getVendedor();?>">
                <p id="centrado">
                    <input name="submit" type="submit"
                        value="Resumen de Venta">
                </p>
            </form>
        </section>
    <footer>
            <h5 id="centrado">Todos los derechos reservados @2015
                        Diseñado por M@nuel Torres</h5>
      </footer>
    </body>
</html>
```

Archivo: **capturaDatos.php**

```
<?php
    //Implementación de funciones de captura de valor
    function getVendedor(){
        return $_POST['txtVendedor'];
    }

    function getNumero(){
        return $_POST['txtN'];
    }

    function getVendedorOculto(){
        return $_POST['txtVendedorOculto'];
    }

    function getNumeroOculto(){
        return $_POST['txtNumeroOculto'];
    }

    function getMonto(){
        return $_POST['txtMonto'];
    }
?>
```

Sólo fines educativos - FreeLibros

Archivo: **resumen.php**

```
<!DOCTYPE html>
<html>
    <head>
        <meta charset="UTF-8">
        <title>Resumen de Facturación</title>
        <link href="estilo.css" rel="stylesheet">
    </head>
    <body>
        <header>
            <h2 id="centrado">Resumen de Registros</h2>
            <p id="centrado">Facturadas registradas por vendedor</p>
            <img src="bannerVendedor.jpg" width="750" height="180"/>
        </header>
    <section>
            <?php require 'capturaDatos.php';?>
            <p id="centrado">EMPLEADO:<?php echo getVendedorOculto(); ?></p>
            <form name="frmResumen" method="POST" action="resumen.php">
            <p>
                <?php
                    $monto = getMonto();
                    foreach($monto as $acum){
                            $s=$s+$acum;
                    }
                    $comision=$s*0.15;
                ?>
            </p>
            <table width="350" border="1"
                   cellspacing="0" cellpadding="5">
                <tr>
                    <td width="152">Total de Venta S/.</td>
                    <td width="135">
                        <?php echo 'S/. '.number_format($s,2,'.',''); ?>
                    </td>
                </tr>
                <tr>
                    <td>15% Comision S/.</td>
                    <td>
                      <?php echo 'S/. '.number_format($comision,2,'.',''); ?>
                    </td>
                </tr>
            </table>
            <p id="centrado">
                <a href="index.php">Retornar</a>
            </p>
        </form>
    </section>
    </body>
</html>
```

Sólo fines educativos - FreeLibros

Sólo fines educativos - FreeLibros

CAP.

8

Archivos

Sólo fines educativos - FreeLibros

Sólo fines educativos - FreeLibros

Cuando declaramos una variable podemos almacenar un valor en la memoria de la computadora, pero al terminar la aplicación esta queda vacía; en un arreglo de elementos ocurre algo parecido pero en grandes cantidades. Es decir, la memoria almacenará varios registros pero igualmente se eliminarán al salir de la aplicación. Una forma de almacenar información permanente es usando los archivos, estos permitirán registrar información y administrarla del modo en que el programador crea conveniente.

Podríamos nombrar algunos casos en los que podamos usar los archivos, por ejemplo, en un contador de visitas donde se almacene un número entero en un archivo, y que aumente cuando un usuario visite el documento web. De la misma manera podríamos trabajar para registrar la venta de un producto, almacenando la fecha y hora en la que se dio la misma. Así PHP ofrece funciones que permiten administrar desde la creación del archivo hasta la manipulación de la data.

8.1 MANEJO DE ARCHIVOS

Veamos las principales funciones con las que cuenta PHP para el manejo y manipulación de los datos de un archivo.

8.1.1 Función file_exist

Permite comprobar si un archivo o carpeta existe en una dirección especificada, su formato es:

```
file_exist('nombre_archivo');
file_exist('nombre_carpeta');
```

Veamos algunos casos:

- Comprobar la existencia del archivo **claves.txt** en la carpeta actual de la aplicación web; en caso exista, emitir el mensaje «Archivo existe..!!», de lo contrario mostrar el mensaje «Archivo no encontrado..!!»

```
<?php
   //Declarando el nombre del archivo a buscar
   $archivo = 'claves.txt';

  //Buscando el archivo
  if (file_exists($archivo))
    echo "Archivo existe..!!";
    else
    echo "Archivo no encontrado";
?>
```

- Comprobar la existencia del archivo **claves.txt**, que se encuentra en la carpeta base; en caso exista, emitir el mensaje «Archivo existe..!!», de lo contrario mostrar el mensaje «Archivo no encontrado..!!»

Sólo fines educativos - FreeLibros

```
<?php
  //Declarando el nombre del archivo a buscar
  $archivo = 'base/claves.txt';

  //Buscando el archivo
  if (file_exists($archivo))
    echo "Archivo existe..!!";
    else
    echo "Archivo no encontrado";
?>
```

8.1.2 Función fopen

Permite abrir un archivo, su formato es:

```
fopen($archivo,modo_de_acceso);
```

Modo	Descripción
R	Indica una apertura de archivo de «solo lectura»; es decir, solo podrá acceder a la información pero no podrá modificar su contenido. Al ejecutar la instrucción lo ubicará en el inicio del archivo.
R+	Indica la apertura del archivo en lectura y escritura; es decir, podría leer y modificar la información contenida. Al ejecutar la instrucción lo ubicará al inicio del archivo.
W	Indica la apertura del archivo «solo para escritura»; es decir, solo para grabar o actualizar información contenida, además de ubicarlo al principio del archivo, bloquea al mismo con una longitud cero; sino no encuentra el archivo, lo crea sin contenido.
W+	Indica la apertura del archivo para lectura y escritura; además de ubicarlo al principio de dicho archivo, lo bloquea con una longitud cero; sino no encuentra al archivo, lo crea sin contenido.
A	Indica la apertura del archivo solo para escritura; es decir, solo para grabar o actualizar, además de ubicar el puntero al final del archivo. Sino encuentra el archivo, lo crea sin contenido.
A+	Indica la apertura del archivo para lectura y escritura, además de ubicar el puntero al final del archivo; sino encuentra el archivo, lo crea sin contenido.

Veamos el siguiente caso:

- Mostrar el contenido del archivo **tablas.txt**, comprobar la existencia del mismo en la carpeta actual de la aplicación web; en caso exista, deberá mostrar su contenido; caso contrario, mostrar el mensaje «Archivo NO existe..!!».

Sólo fines educativos - FreeLibros

```php
<?php
    //Definición del nombre del archivo
    $archivo = "tablas.txt";

    //Verificando la existencia del archivo
    if (!file_exists($archivo)){
        echo 'Archivo NO existe..!!';
    }else{
        //Abriendo en forma de lectura
        $abrir = fopen($archivo, "r");

        //Obtener el contenido a partir de la lectura
        $contenido = fread($abrir, filesize($archivo));

        //Imprimir el contenido del archivo
        echo $contenido;
    }
?>
```

8.1.3 Función fclose

Permite el cierre de un determinado archivo previamente abierto, habitualmente se cierra un archivo cuando posiblemente se vuelva a usar, su formato es:

```php
fclose($archivo_abierto);
```

Veamos el siguiente caso:

- Mostrar el contenido del archivo **tablas.txt**, comprobar la existencia del mismo en la carpeta actual de la aplicación web. En caso existe, mostrar su contenido y cerrar el archivo; caso contrario, mostrar el mensaje «Archivo NO existe..!!»

```php
<?php
    //Definición del nombre del archivo
    $archivo = "tablas.txt";

    //Verificando la existencia del archivo
    if (!file_exists($archivo)){
        echo 'Archivo NO existe..!!';
    }else{
        //Abriendo en forma de lectura
        $abrir = fopen($archivo, "r");

        //Obtener el contenido a partir de la lectura
        $contenido = fread($abrir, filesize($archivo));

        //Cerrando el archivo
        fclose($abrir);

        //Imprimir el contenido del archivo
        echo $contenido;
    }
?>
```

Sólo fines educativos - FreeLibros

8.1.4 Función fwrite

Permite escribir dentro de un archivo previamente abierto, su formato es:

```
fwrite($archivo_abierto, 'elemento');
```

Veamos el siguiente caso:

- Registrar los siguientes productos dentro del archivo tablas.txt:

Lavadora industrial	1500	20
Televisión 40 pulgadas	3500	30
Refrigeradora 12 pies	5500	10

```
<?php
 //Abrir el archivo
 $archivo = 'tablas.txt';
 $abrir = fopen($archivo, 'w');

 //Información de los productos
 //          Descripción           precio stock
 $producto1 = 'Lavadora industrial-1500-20'.chr(13).chr(10);
 $producto2 = 'Televisión 40 pulgadas-3500-30'.chr(13).chr(10);
 $producto3 = 'Refrigeradora 12 pies - 5500 - 10';

 //Enviar información de los productos
 fwrite($abrir, $producto1);
 fwrite($abrir, $producto2);
 fwrite($abrir, $producto3);

 //Cerrar el archivo
 fclose($abrir);
?>
```

8.1.5 Función fread

Permite leer un determinado archivo previamente abierto, hasta la especificación de su longitud, su formato es:

```
fread($archivo_abierto,longitud_del_archivo);
```

Hay que tener en cuenta que el **archivo_abierto** usa la función **fopen**, y la longitud del archivo se mide con la función **filesize**.

Veamos los siguientes casos:

- Mostrar el contenido del archivo **tablas.txt**, usando la función **readfile**.

```
<?php
    //Nombre del archivo
    $filename = "tablas.txt";

    //Leer y mostrar el contenido
    readfile($filename);
?>
```

Sólo fines educativos - FreeLibros

El resultado es:

```
Lavadora industrial - 1500 - 20 Televisión 40 pulgadas - 3500 - 30 Refrigeradora 12 pies - 5500 - 10
```

La función **readfile** se diferencia de **fread** en que no necesita aplicar la función **fopen** al archivo.

- Mostrar el contenido del archivo **tablas.txt**, de tal forma que los productos se visualicen uno debajo del otro.

```
<?php
    //Especificar el nombre del archivo
    $productos = fopen('tablas.txt','r');

    //Aperturar el archivo
    $leer = fread($productos,  filesize('tablas.txt'));

    //Obtener líneas de registros de productos
    $linea = explode(chr(13).chr(10),$leer);

    //Imprimir todos los productos
    for ($i=0;$i<count($linea);$i++){
        $palabra = explode('-',$linea[$i]);
        echo $palabra[0];
        echo $palabra[1];
        echo $palabra[2].'<br>';
    }
?>
```

El resultado es:

```
Lavadora industrial 1500 20
Televisión 40 pulgadas 3500 30
Refrigeradora 12 pies 5500 10
```

8.1.6 Función fgets

Permite leer y obtener línea por línea desde un archivo previamente abierto, su formato es:

```
fgets($archivo_abierto,longitud_del_archivo);
```

Hay que tener en cuenta que el **archivo_abierto** usa la función **fopen**, y la longitud del archivo se mide con la función **filesize**.

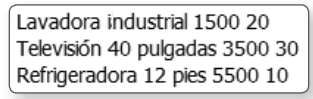
Veamos los siguientes casos:

- Mostrar el contenido del archivo tablas.txt, usando la función **Fgets**.

Sólo fines educativos - FreeLibros

```php
<?php
  //Especificar el archivo de texto
  $archivo = 'tablas.txt';

  //Abrir el archivo
  $abrir = fopen($archivo, 'r');

  //Imprimir el registro de los productos
  echo 'Los productos son: <br>';
  while (($producto=fgets($abrir,filesize($archivo)))!==false)
  {
        echo $producto.'<br>';
  }
  fclose($abrir);
?>
```

El resultado es:



8.1.7 Función fputs

Tiene la misma funcionalidad que la instrucción **fwrite**, su formato es:

```
fputs($archivo_abierto, 'elemento')
```

Veamos el siguiente caso:

- Mostrar el contenido del archivo **tablas.txt**, de tal forma que los productos se visualicen uno debajo del otro.

```php
<?php
 //Abrir el archivo
 $archivo = 'tablas.txt';
 $abrir = fopen($archivo, 'a+');

 //Información de los productos
 //          Descripción           precio stock
 $producto1 = 'Lavadora Industrial-1500-20'.chr(13).chr(10);
 $producto2 = 'Televisión 40 pulgadas-3500-30'.chr(13).chr(10);
 $producto3 = 'Refrigeradora 12 pies - 5500 - 10';

 //Enviar información de los productos
 fputs($abrir, $producto1);
 fputs($abrir, $producto2);
 fputs($abrir, $producto3);

 //Cerrar el archivo
 fclose($abrir);
?>
```

Sólo fines educativos - FreeLibros

8.1.8 Función rewind

Permite reinicializar la posición del puntero dentro de un archivo; es decir, se asignará a la posición inicial el puntero, su formato es:

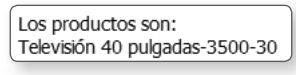
```
rewind($archivo_abierto);
```

Veamos el siguiente caso:

- Renombraremos el siguiente producto 'Lavadora Industrial-1500-20' por el siguiente 'Televisión 40 pulgadas-3500-30' usando la función **rewind**.

```
<?php
 //Seleccionando el archivo
 $archivo = 'tablas.txt';

 //Apertura del archivo para leer y escribir
 $abrir = fopen($archivo, 'r+');

 //Registrando el primero producto
 $descripcion='Lavadora Industrial-1500-20'.chr(13).chr(10);
 fwrite($abrir, $descripcion);

 //Ubicando el puntero al principio del archivo
 rewind($abrir);

 //Registrando el segundo producto
 $descripcion='Televisión 40 pulgadas-3500-30'.chr(13).chr(10);
 fwrite($abrir, $descripcion);

 //Ubicando el puntero al principio del archivo
 rewind($abrir);

 //Imprimir el registro de los productos
 echo 'Los productos son: <br>';
 while (($producto=fgets($abrir,filesize($archivo)))!==false)
 {
        echo $producto.'<br>';
 }

 //Cerrando el archivo
 fclose($abrir);
?>
```

El resultado es:



8.1.9 Función filectime

Permite obtener la fecha del último cambio realizado en un archivo, su formato es:

```
filectime($archivo);
```

Veamos el siguiente caso:

- Renombraremos el siguiente producto 'Lavadora Industrial-1500-20' por el siguiente 'Televisión 40 pulgadas-3500-30' usando la función **rewind**.

Sólo fines educativos - FreeLibros

```php
<?php
    //Especificación del nombre del archivo
    $archivo = 'productos.txt';

    //Verificando la existencia del archivo
    if (file_exists($archivo)) {
        echo "La última modificación de $archivo fue: " .
                date("F d Y H:i:s.", filectime($archivo));
    }
?>
```

8.1.10 Función file

Permite transferir todo la información contenida en un archivo en un arreglo, su formato es:

```php
$arreglo = file($archivo);
```

Veamos el siguiente caso:

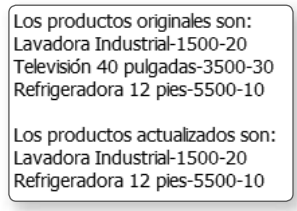
- De una lista de productos en el archivo **tablas.txt** elimine el producto que se encuentra en la posición 2, debe asumir los siguientes productos en el archivo **tablas.txt**:

Lavadora industrial	1500	20
Televisión 40 pulgadas	3500	30
Refrigeradora 12 pies	5500	10

```php
<?php
    //Seleccionando el archivo
    $archivo = "tablas.txt";
    echo '<br>Los productos originales son: <br>';
    listar($archivo);

    //Crear un arreglo de los productos
    $producto = file($archivo);

    //Eliminando el segundo producto
    $posición = 2;
    unset ($producto[$posicion-1]);

    //Reinicializando el arreglo
    $data = array_values($producto);

    //Sobrescribir en el archivo
    file_put_contents($archivo, implode($producto));

    //Imprimir los productos actualizados
    echo '<br>Los productos actualizados son: <br>';
    listar($archivo);

    function listar($archivo){
    $abrir = fopen($archivo, 'r');
    while(($producto=fgets($abrir,filesize($archivo)))!==false)
    {
        echo $producto.'<br>';
    }
    fclose($abrir);
    }
?>
```

Sólo fines educativos - FreeLibros

El resultado es:



8.1.11 Función file_put_contents

Permite añadir información a un archivo, su formato es:

```
file_put_contents($archivo, 'elemento_a_agregar');
```

Veamos el siguiente caso:

- De una lista de productos en el archivo **tablas.txt**, debemos agregar un nuevo producto al final de los registros; debe asumir los siguientes productos en el archivo **tablas.txt**:

Lavadora industrial	1500	20
Refrigeradora 12 pies	5500	10

```
<?php
    //Listado los productos originales
    $archivo = 'tablas.txt';
    echo 'Los productos originales son: <br>';
    listar($archivo);

    // Abre el fichero para obtener el contenido existente
    $actual = file_get_contents($archivo);

    // Añade una nueva persona al fichero
    $actual .= "Radiograbadora-500-40\n";

    // Escribe el contenido al fichero
    file_put_contents($archivo, $actual);

    //Lista los productos actualizados
    echo '<br>Los productos actualizados son: <br>';
    listar($archivo);

  function listar($archivo){
   $abrir = fopen($archivo, 'r');
   while(($producto=fgets($abrir,filesize($archivo)))!==false)
   {
        echo $producto.'<br>';
   }
   fclose($abrir);
  }
?>
```

Sólo fines educativos - FreeLibros

El resultado es:

```
Los productos originales son:
Lavadora industrial-1500-20
Refrigeradora 12 pies-5500-10

Los productos actualizados son:
Lavadora industrial-1500-20
Refrigeradora 12 pies-5500-10
Radiograbadora-500-40
```

8.2 Manejo de archivos y carpetas

PHP puede controlar también los directorios, ahora veremos algunas funciones básicas, como recorrer por todos los registros, eliminar o modificar el nombre de un archivo.

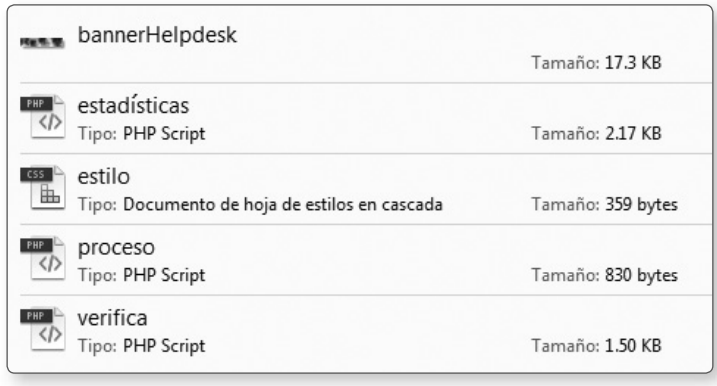
8.2.1 Función scandir

Permite obtener los archivos contenidos en una determinada carpeta, su formato es:

```
scandir($ruta_carpeta);
```

Veamos el siguiente caso:

- Listaremos los archivos y sus tamaños en *kilobytes* contenidos en la carpeta **Estadísticas**, ubicada en el servidor; la imagen muestra el contenido de la carpeta:




```
<?php
  //Especificar la carpeta de donde se analizara su contenido
  $carpeta = './estadisticas';

  //Obtener sus archivos
  $archivos = scandir($carpeta);

  //Listar los archivos con sus respectivos tamaños
  foreach ($archivos as $campo) {
    if (is_file("$carpeta/$campo") &&
                $campo != '.' && $campo != '..') {

       echo "$campo - ".filesize("$carpeta/$campo")."kb<br>";
    }
  }
?>
```

Sólo fines educativos - FreeLibros

El resultado es:

```
bannerHelpdesk.jpg - 17782kb
estadisticas.php - 2224kb
estilo.css - 359kb
proceso.php - 830kb
verifica.php - 1546kb
```

8.2.2 Función unlink

Permite eliminar un determinar archivo, su formato es:

```
unlink($archivo);
```

Veamos el siguiente caso:

- Eliminaremos el archivo **reloj.jpg** que se encuentra en la carpeta actual:

```
<?php
    //Especificando el archivo a eliminar
    $archivo = "reloj.jpg";

    //Verificando si el archivo existe
    if (file_exists($archivo)) {
        unlink ($archivo);
        echo 'El archivo se elimino correctamente..!!';
        } else {
        echo 'El archivo: '.$archivo.' no existe';
    }
?>
```

8.2.3 Función rename

Permite modificar el nombre de un determinado archivo, su formato es:

```
rename($nombre_actual,$nuevo_nuevo);
```

Veamos el siguiente caso:

- Modificaremos el nombre del archivo **tablas.txt** por el nombre **productos.txt**:

```
<?php
  //Especificando el nombre del archivo actual
  $original = 'tablas.txt';

  //Especificando el nuevo nombre del archivo
  $nuevo = 'productos.txt';

  //Verificando la existencia del archivo
  if (file_exists($original)) {
  rename ($original, $nuevo);
  echo 'Archivo '.$original.'eliminado correctamente..!!';
  } else {
  echo 'Ocurrió un problema al cambiar el nombre de'.$original;
  }
?>
```

Sólo fines educativos - FreeLibros

8.3 Casos desarrollados

Para la solución de los casos desarrollados se recomienda crear un directorio o carpeta dentro del servidor local.

Caso desarrollado 1: Verificar la existencia de un archivo

Implemente una aplicación web con PHP que permita ingresar el nombre de un archivo, y evalúe si el archivo existe, cual es la última fecha de modificación, qué tipo de archivo es, qué tamaño tiene y si el archivo es ejecutable o no.

La interfaz gráfica inicial es la siguiente:

Tener en cuenta:

- Al presionar el botón **Verificar** se mostrarán los datos solicitados por el caso.
- Implemente una función que capture del nombre del archivo llamado **getArchivo**.
- Implemente una función que verifique si el archivo ingresado es correcto o no, muestre un mensaje si hay un error.
- Implemente funciones que permitan obtener los datos solicitados en el caso, como fecha de la modificación del archivo (**ultimaModificacion**), tipo de archivo, tamaño del archivo y para determinar si es ejecutable.
- Contaremos con dos archivos para la aplicación, la primera será **index.html**, que contendrá el entorno de la aplicación; habrá otra llamada **proceso.php**, que permitirá implementar todas las funciones necesarias para la aplicación.
- Solo se debe mostrar la información de las estadísticas cuando se presione el botón **Verificar**.
- Finalmente, el resultado se debe mostrar como la siguiente interfaz:

Sólo fines educativos - FreeLibros

Archivo: **index.php**

```
<!DOCTYPE html>
<html>
    <head>
        <meta charset="UTF-8">
        <title>Verificar Existencia de archivo</title>
        <link href="estilo.css" rel="stylesheet">
    </head>
    <body>
        <header>
            <h2 id="centrado">
                VERIFICAR LA EXISTENCIA DE UN ARCHIVO
            </h2>
            <img src="bannerHelpdesk.jpg" width="550" height="155"/>
        </header>
        <section>
            <?php
                error_reporting(0);
                require 'proceso.php';
            ?>
            <form name="frmArchivo" method="POST">
                <table border="1" width="550" cellspacing="1" cellpadding="1">
                <tr>
                    <td>Nombre del Archivo</td>
                    <td><input type="text" name="txtArchivo"
                               value="" /></td>
                </tr>
                <tr>
                    <td></td>
                    <td><input type="submit" value="Verificar"
                               name="btnVerificar" /></td>
                </tr>
                <?php if (isset($_POST['btnVerificar'])) {?>
                <tr>
                    <td>Nombre completo del archivo</td>
                    <td><?php echo getArchivo(); ?></td>
                </tr>
```

Sólo fines educativos - FreeLibros

```
                <tr>
                    <td>¿Existe el archivo?</td>
                    <td><?php echo verifica(getArchivo()); ?></td>
                </tr>
                <tr>
                    <td>Fecha de la ultima modificación</td>
                    <td><?php echo ultimaModificacion(getArchivo()); ?></td>
                </tr>
                <tr>
                    <td>Tipo de archivo</td>
                    <td><?php echo tipoArchivo(getArchivo()); ?></td>
                </tr>
                <tr>
                   <td>Tamaño en bytes</td>
                   <td><?php echo tamañoArchivo(getArchivo()).'Bytes'; ?></td>
                </tr>
                <tr>
                    <td>¿El archivo es ejecutable?</td>
                    <td><?php echo esEjecutable(getArchivo()); ?></td>
                </tr>
                <?php } ?>
                </table>
            </form>
        </section>
        <footer>
            <h6 id="centrado">Todos los derechos reservados –
                Lic. Manuel Torres</h6>
        </footer>
    </body>
</html>
```

Archivo: **proceso.php**

```
<?php
    function getArchivo(){
        return $_POST['txtArchivo'];
    }

    function verifica($archivo){
        if (file_exists($archivo))
            return 'Es correcto el archivo..!!';
        else
            return 'El archivo NO existe..!!';
    }

    function ultimaModificacion($archivo){
        if (file_exists($archivo))
            return date("F d Y H:i:s.", filectime($archivo));
        else
            return '';
    }

    function tipoArchivo($archivo){
        return filetype($archivo);
    }

    function tamañoArchivo($archivo){
        return filesize($archivo);
    }
```

Sólo fines educativos - FreeLibros

```
    function esEjecutable($archivo){
        if (is_executable($archivo))
            return 'Archivo ejecutable';
        else
            return 'Archivo NO ejecutable';
    }
?>
```

Comentarios:

```
<?php
   error_reporting(0);
   require 'proceso.php';
?>
```

Omitimos los errores de advertencia de PHP, luego invocamos a todos los métodos implementados en el archivo **proceso.php**, usando la función **require**; debemos considerar que la invocación al archivo puede realizarse con o sin paréntesis, pero siempre debe ir encerrado entre comillas simples o dobles.

```
<tr>
    <td>Nombre completo del archivo</td>
    <td><?php echo getArchivo(); ?></td>
</tr>
```

El método **getArchivo** obtiene el nombre del archivo ingresado por el usuario en el control **txtArchivo**, y este valor es impreso en el mismo control para que no desaparezca luego de presionar el botón **Verificar**.

```
function getArchivo(){
        return $_POST['txtArchivo'];
}
```

Función que permite obtener el nombre del archivo ingresado por el usuario desde el control **txtArchivo**.

```
function verifica($archivo){
        if (file_exists($archivo))
            return 'Es correcto el archivo..!!';
        else
            return 'El archivo NO existe..!!';
}
```

Función que permite verificar si el archivo existe en la carpeta actual del proyecto, la función **file_exists** emite como resultado **true** cuando existe el archivo; en nuestro caso, imprimiremos el texto «Es correcto el archivo..!!»; y **false** cuando no lo encuentra, en cuyo caso se mostrará el mensaje «El archivo NO existe..!!».

Sólo fines educativos - FreeLibros

```
function ultimaModificacion($archivo){
        if (file_exists($archivo))
            return date("F d Y H:i:s.", filectime($archivo));
        else
            return '';
}
```

Función que permite devolver la fecha de la última modificación realizada en un archivo determinado, para este caso enviamos como parámetro el nombre del archivo registrado por el usuario; la función **filectime** obtiene el momento de la última modificación en un archivo.

```
function tipoArchivo($archivo){
        return filetype($archivo);
}
```

Función que permite devolver el tipo de archivo a partir de la función **filetype**.

```
function tamañoArchivo($archivo){
        return filesize($archivo);
}
```

Función que permite devolver el tamaño en *bytes* del contenido de un archivo determinado.

```
function esEjecutable($archivo){
        if (is_executable($archivo))
            return 'Archivo ejecutable';
        else
            return 'Archivo NO ejecutable';
}
```

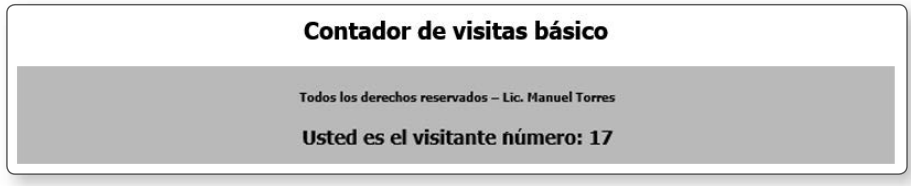
Función que determina si un archivo es de tipo ejecutable o no, para lo cual solo emitimos mensajes a partir de la evaluación con la función **is_executable**.

Caso desarrollado 2: Contador de visita básico

Implemente una aplicación web con PHP que permita mostrar el número de visita que un usuario realiza sobre nuestra aplicación.

Tener en cuenta:

- Implementar toda la aplicación en una carpeta llamada **ContadorSimple**.
- Debemos tener un archivo de texto llamado **contador.txt** dentro del cual debe almacenarse el número de visita y que se actualizará cada vez que presione F5 sobre la aplicación.
- Inicialmente el archivo **contardor.txt** debe contener el número 0.
- Finalmente, el resultado se debe mostrar como la siguiente interfaz:



Sólo fines educativos - FreeLibros

Código: **index.php**

```
<!DOCTYPE html>
<html>
    <head>
        <meta charset="UTF-8">
        <title>Contador de visitas básico</title>
        <link href="estilo.css" rel="stylesheet">
    </head>
    <body>
        <header>
            <h2 id="centrado">Contador de visitas básico</h2>
        </header>
        <section>
        </section>

        <footer>
            <table width="800">
               <tr>
                 <th>
                   <h6 id="centrado">Todos los derechos reservados –
                                                Lic. Manuel Torres
                   </h6>
                   <p id="centrado">
                    <?php
                        $visita = "contador.txt";
                        include('cuenta.php');
                        contador($visita);
                     ?>
                    </p>
                 </th>
               </tr>
            </table>
        </footer>
    </body>
</html>
```

Código: **cuenta.php**

```
<?php
    function contador($archivo){
        $fp = fopen($archivo, 'rw');

        $num=  fgets($fp,5);
        $num+=1;
        echo 'Usted es el visitante numero: '.$num;
    }
?>
```

Comentarios:

```
<?php
  $visita = "contador.txt";
  include('cuenta.php');
  contador($visita);
?>
```

La variable **$visita** registra el nombre del archivo que se quiere evaluar, para lo cual dicho archivo debe encontrarse en la carpeta del archivo actual con el número cero como contenido. Invocamos al archivo **cuenta.php** que contiene el método contador. Es así que en la tercera línea podemos invocar a la función contador, enviándole como parámetro el nombre del archivo a evaluar.

Sólo fines educativos - FreeLibros

```
function contador($archivo){
  $fp = fopen($archivo, 'rw');

  $num=  fgets($fp,5);
  $num+=1;
  echo 'Usted es el visitante numero: '.$num;
}
```

La función contador tiene por misión aumentar en uno el contenido del archivo **contador.txt**. Para lograr esto, primero debemos abrir el archivo con la función **fopen** con la opción **rw** de lectura y escritura: lectura para obtener el valor, escritura para actualizar el número de visitas.

La variable **$num** almacena el valor encontrado dentro del archivo, esto está representado por la variable **$fp**, el valor 5 representa el máximo tamaño a obtener desde el archivo, en algunos casos se puede especificar cifras mayores para contenidos extensos en el archivo. Al valor obtenido en la variable **$num** se le aumenta en uno y es impreso para que el usuario observe el número de visitas que generó.

Caso desarrollado 3: Contador de visitas de forma gráfica

Implemente una aplicación web con PHP que permita mostrar el número de visita de forma gráfica, esta se actualizará cuando el usuario presione F5.

Tener en cuenta:

- Implementar toda la aplicación en una carpeta llamada **ContadorGrafico**.
- Tener una carpeta con las imágenes de los números, tal como se muestra a continuación:

- Debemos tener un archivo de texto llamado **contador.txt**, dentro del cual se almacenará el número de visita; inicialmente el archivo debe contener el número 0.
- Finalmente, el resultado se debe mostrar como la siguiente interfaz:

Sólo fines educativos - FreeLibros

Archivo: **index.php**

```
<!DOCTYPE html>
<html>
    <head>
        <meta charset="UTF-8">
        <title>Contador de visitas con imágenes</title>
        <link href="miEstilo.css" rel="stylesheet">
    </head>
    <body>
        <header>
            <h2 id="centrado">Contador de visitas con imágenes</h2>
        </header>
        <section>
        </section>

        <footer>
            <table width="800">
               <tr>
                   <th>
                       <h6 id="centrado">Todos los derechos reservados –
                                                Lic. Manuel Torres
                        </h6>
                        <p>
                        <?php
                            error_reporting(0);
                            include('cuenta.php');
                            echo 'Usted es el visitante Nº '.numero();
                        ?>
                        </p>
                    </th>
                </tr>
            </table>
        </footer>

    </body>
</html>
```

Archivo: **cuenta.php**

```
<?php
function numero(){
    $archivo = 'contador.txt';
    $abrir = fopen($archivo, 'r+');

    $numero= fread ($abrir, filesize($archivo));

    rewind($abrir);

    $numero+=1;
    fputs($abrir, $numero);
    fclose($abrir);

    for($i=0; $i<strlen($numero);$i++){
        $s=substr($numero, $i,1);
        $img.='<img src="numeros/'.$s.'.gif"/>';
    }
    return $img;
}
?>
```

Sólo fines educativos - FreeLibros

Archivo: **miestilo.css**

```
body{
    font-family: tahoma;
    font-size: 14px;
}
#centrado{
    text-align: center;
}
table {
    margin: auto;
}
table th {
background-color: #b9c9fe;
}
```

Comentarios:

```
<?php
   error_reporting(0);
   include('cuenta.php');
   echo 'Usted es el visitante Nº '.numero();
?>
```

Omitimos los errores de advertencia de PHP con la función **error_reporting(0)**, incluimos el archivo **cuenta.php**, el cual contiene la función **numero** que permite imprimir el número de visita correspondiente.

```
function numero(){
    $archivo = 'contador.txt';
    $abrir = fopen($archivo, 'r+');

    $numero= fread ($abrir, filesize($archivo));

    rewind($abrir);

    $numero+=1;
    fputs($abrir, $numero);
    fclose($abrir);

    for($i=0; $i<strlen($numero);$i++){
        $s=substr($numero, $i,1);
        $img.='<img src="numeros/'.$s.'.gif"/>';
    }
    return $img;
}
```

La función número permite abrir el archivo **contador.txt**, que contiene el número de visita actualizado; primero, abrimos el archivo para obtener su contenido con la función **fopen**, la variable **$numero** obtiene el número almacenado en el archivo **contador.txt**; la función **rewind** ubica el puntero en el valor inicial de un archivo abierto.

La variable **$numero** aumenta en uno para poder actualizar el contador contenido en el archivo **contador.txt**, que luego será registrado en dicho archivo con la función **fputs**. Considere que esta función es similar a la función **fwrite**; finalmente, se cierra el archivo con la función **fclose**; esta actividad es obligatoria pues un archivo abierto siempre tiene que cerrarse.

La estructura **for** tiene por misión determinar la cantidad de dígitos que contiene el archivo para poder imprimir las imágenes en pareja, y así obtener correctamente los números de dos, tres y cuatro cifras.

Sólo fines educativos - FreeLibros

Caso desarrollado 4: Control de registro de clientes

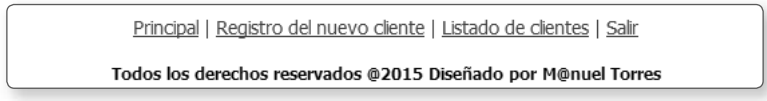
Implemente una aplicación web con PHP que permita tener el control de los registros de los clientes, para lo cual deberá solicitar los datos, como nombres completos, dirección, teléfono y fecha de nacimiento.

Tener en cuenta:

- Implementar toda la aplicación en una carpeta llamada **RegistroClientes**.
- Todo registro de clientes debe guardarse en un archivo llamado **clientes.txt**.
- Deberá definir los siguientes archivos:

  - **cabecera.php**: Contiene la cabecera que se repetirá a lo largo de las páginas web del proyecto.

  - **pie.php**: Contiene el pie de página que se repetirá a lo largo de las páginas web del proyecto.



  - **index.php**: Contiene la página principal del proyecto e incluye los archivos **encabezado.php** y **pie.php**.

Sólo fines educativos - FreeLibros

Las imágenes usadas tienen el siguiente nombre:

icono_cliente.png

icono_lista.png

icono_salida.png

- Al seleccionar **Registro del nuevo cliente** deberá mostrar la siguiente pantalla:

Aquí debemos tener en cuenta que el número de registro debe ser automático y que al ingresar todos los valores debe presionar el botón **Registrar cliente**; si todo es correcto se retornará a la ventana principal de forma automática, recuerde que cada vez que inicia este archivo debe generarse automáticamente el número de registro.

- Finalmente, al seleccionar **Listado de clientes**, se debe mostrar como la siguiente interfaz:

Al iniciar la página se deben mostrar todos los registros de clientes almacenados en el archivo **clientes.txt**.

Sólo fines educativos - FreeLibros

Archivo: **index.php**

```
<!DOCTYPE html>
<html>
    <head>
        <meta charset="UTF-8">
        <title>CONTROL DE CLIENTES</title>
        <link href="estilo.css" rel="stylesheet">
    </head>
    <body>
        <header>
            <?php require 'encabezado.php'; ?>
        </header>
        <section>
            <table border="1" width="550" cellspacing="5">
            <tr>
                <td><a href="registro.php">
                          <img src="icono_cliente.png"
                               width="60" height="60" /></a>
                </td>
                <td><a href="registro.php">Registro de clientes</a></td>
            </tr>
            <tr>
                <td><a href="listado.php">
                          <img src="icono_lista.png"
                               width="60" height="60" /></a>
                </td>
                <td><a href="listado.php">Listado de Clientes</a></td>
            </tr>
            <tr>
                <td><a href="javascript:close()">
                         <img src="icono_salida.png"
                              width="60" height="60" /></a>
                </td>
                <td><a href="javascript:close()">Salir</a></td>
            </tr>
            </table>
        </section>
        <footer>
            <?php require 'pie.php'; ?>
        </footer>
    </body>
</html>
```

Archivo: **registro.php**

```
<!DOCTYPE html>
<html>
    <head>
        <meta charset="UTF-8">
        <title>Registro de Clientes</title>
        <link href="estilo.css" rel="stylesheet">
    </head>
    <body>
        <header>
            <?php require 'encabezado.php';?>
            <h2 id="centrado">REGISTRO DEL NUEVO CLIENTE</h2>
        </header>
        <section>
            <form name="frmRegistro" method="POST">
            <table border="1" width="550" cellspacing="5" cellpadding="0">
```

Sólo fines educativos - FreeLibros

```
                <tr>
                    <td>Nº REGISTRO</td>
                    <td>
                        <?php
                            error_reporting(0);
                            require 'generaCodigo.php';
                            $clientes = 'clientes.txt';
                            $numero = contador($clientes);
                            echo $numero;
                        ?>
                    </td>
                </tr>
                <tr>
                    <td>CLIENTE</td>
                    <td><input type="text" name="txtCliente" size="60" /></td>
                </tr>
                <tr>
                  <td>DIRECCION</td>
                  <td><input type="text" name="txtDireccion" size="60" /></td>
                </tr>
                <tr>
                    <td>TELEFONO</td>
                    <td><input type="text" name="txtFono" size="20" /></td>
                </tr>
                <tr>
                    <td>FECHA NAC.</td>
                    <td><input type="text" name="txtFecha" size="20" /></td>
                </tr>
                <tr>
                    <td></td>
                    <td><input type="submit"
                               value="REGISTRAR CLIENTE"
                               name="btnRegistrar" />
                    </td>
                </tr>
                <tr>
                    <td></td>
                    <td><?php
                            if(isset($_POST["btnRegistrar"])){
                                include('captura.php');
                                include('grabar.php');
                                registra($numero, getCliente(),
                                         getDireccion(),
                                         getTelefono(), getFecha());
                                header('location:index.php');
                            }
                        ?>
                    </td>
                </tr>
            </table>
        </form>
        </section>
        <footer>
           <?php require 'pie.php'; ?>
        </footer>
    </body>
</html>
```

Sólo fines educativos - FreeLibros

Archivo: **grabar.php**

```
<?php
    function registra($cod,$cli,$dir,$tel,$fec){
        $archivo = fopen('clientes.txt','a+');
        $unCliente = $cod.'|'.$cli.'|'.$dir.'|'.$tel.'|'.$fec.chr(13).chr(10);

        fwrite($archivo, $unCliente);
        fclose($archivo);
        header('index.php');
    }
?>
```

Archivo: **listado.php**

```
<!DOCTYPE html>
<html>
    <head>
        <meta charset="UTF-8">
        <title>Listado de Clientes</title>
        <link href="estilo.css" rel="stylesheet">
    </head>
    <body>
        <header>
            <?php require 'encabezado.php';?>
            <h2 id="centrado">LISTADO DE CLIENTES</h2>
        </header>
        <section>
            <table border="1" width="550" cellspacing="0" cellpading="0">
            <tr>
                <td>CODIGO</td>
                <td>NOMBRE DEL CLIENTE</td>
                <td>DIRECCION</td>
                <td>TELEFONO</td>
                <td>FECHA NACIMIENTO</td>
            </tr>
            <tr>
                <?php
                    $clientes = fopen('clientes.txt','r');
                    if(filesize('clientes.txt')==0)
                        echo '<p id="centrado">No hay registros..!!</p>';
                        else {
                          $leer = fread($clientes,  filesize('clientes.txt'));
                          $linea = explode(chr(13).chr(10),$leer);
                          for ($i=0;$i<count($linea)-1;$i++){
                             $palabra = explode('|',$linea[$i]);
                ?>
                             <td><?php echo $palabra[0]; ?></td>
                             <td><?php echo $palabra[1]; ?></td>
                             <td><?php echo $palabra[2]; ?></td>
                             <td><?php echo $palabra[3]; ?></td>
                             <td><?php echo $palabra[4]; ?></td>
            </tr>
                <?php
                    }
                        }
                ?>
            </table>
        </section>
        <footer>
            <?php require 'pie.php'; ?>
        </footer>
    </body>
</html>
```

Sólo fines educativos - FreeLibros

Archivo: **generaCodigo.php**

```
<?php
    function contador($archivo){
        $clientes = fopen($archivo,'r');
        $leer = fread($clientes,  filesize($archivo));
        $linea = explode(chr(13).chr(10),$leer);
        $n = count($linea);
        return $n;
    }
?>
```

Archivo: **encabezado.php**

```
<!DOCTYPE html>
<html>
    <head>
        <meta charset="UTF-8">
        <title></title>
    </head>
    <body>
        <h2 id="centrado">CONTROL DE CLIENTES</h2>
        <img src="banner_clientes.jpg" width="550" height="120" />
    </body>
</html>
```

Archivo: **pie.php**

```
<!DOCTYPE html>
<html>
    <head>
        <meta charset="UTF-8">
        <title></title>
    </head>
    <body>
        <p id="centrado">
            <a href="index.php">Principal</a> |
            <a href="registro.php">Registro del nuevo clientes</a> |
            <a href="listado.php">Listado de clientes</a> |
            <a href="javascript:close()">Salir</a>
        </p>
        <h5 id="centrado">Todos los derechos reservados @2015
                          Diseñado por M@nuel Torres</h5>
    </body>
</html>
```

Archivo: **estilo.css**

```
body{
    font-family: tahoma;
    font-size: 14px;
}
#centrado{
    text-align: center;
}
table {
    margin: auto;
}
img{
    margin: auto;
    display: block;
```

Sólo fines educativos - FreeLibros

```
}
td {
    border: solid 1px #006699;
    border-top: #006699;
    border-right: #006699;
    border-left: #006699;
    border-bottom: #006699;
}
```

Comentarios:

```
<?php
        error_reporting(0);
        require 'generaCodigo.php';
        $clientes = 'clientes.txt';
        $numero = contador($clientes);
        echo $numero;
?>
```

Omitimos los errores de advertencia de PHP, luego incorporamos el archivo **generaCodigo.php**, que contiene la función contador que permitirá determinar cuántos clientes tenemos registrados hasta el momento y así asignar un numero correlativo al registro.

La variable **$clientes** registra el nombre del archivo **clientes.txt**, que contiene todos los registros de los clientes hasta el momento. La variable **$numero** registra el número autogenerado desde la función contador.

```
<?php
    if(isset($_POST["btnRegistrar"])){
        include('captura.php');
        include('grabar.php');
        registra($numero, getCliente(), getDireccion(), getTelefono(), getFecha());
        header('location:index.php');
    }
?>
```

Se incluye el archivo **captura.php**, que permite obtener todos los valores registrados por el usuario por medio de funciones. Luego se incluye el archivo **grabar.php**, que contiene la función **registra**; esta función enviará todos los valores obtenidos al registro de clientes almacenado en el archivo. Finalmente, se direcciona al archivo **index.php** para seguir registrando un nuevo cliente.

```
function registra($cod,$cli,$dir,$tel,$fec){
    $archivo = fopen('clientes.txt','a+');
    $unCliente = $cod.'|'.$cli.'|'.$dir.'|'.$tel.'|'.$fec.chr(13).chr(10);

    fwrite($archivo, $unCliente);
    fclose($archivo);
    header('index.php');
  }
```

La función **registra** permite enviar la información al archivo **clientes.txt**; por tal motivo, esta función debe tener parámetros con todos los valores a registrar. Se abre el archivo **clientes.txt** con la función **fopen**, se crea la variable **$unCliente** para enviar en una sola línea el registro del nuevo cliente añadiendo un cambio de línea al final de cada registro, para lo cual usamos la combinación de funciones **chr(13).chr(10)**.

Sólo fines educativos - FreeLibros

La función **fwrite** permite enviar la información del nuevo cliente al archivo **clientes.txt**, representado por la variable **$archivo**. Se cierra el archivo con la función **fclose** y se direcciona al archivo **index.php** con la función **header**.

```
<tr>
    <?php
      $clientes = fopen('clientes.txt','r');
      if(filesize('clientes.txt')==0)
          echo '<p id="centrado">No hay registros..!!</p>';
          else {
          $leer = fread($clientes,  filesize('clientes.txt'));
          $linea = explode(chr(13).chr(10),$leer);
          for ($i=0;$i<count($linea)-1;$i++){
              $palabra = explode('|',$linea[$i]);
    ?>
              <td><?php echo $palabra[0]; ?></td>
              <td><?php echo $palabra[1]; ?></td>
              <td><?php echo $palabra[2]; ?></td>
              <td><?php echo $palabra[3]; ?></td>
              <td><?php echo $palabra[4]; ?></td>
</tr>
<?php
              }
                }
?>
```

Para poder imprimir el listado de clientes registrados en el archivo **clientes.txt**, debemos abrir el archivo con la función **fopen** de forma de lectura. Luego, validamos el tamaño del archivo **clientes.txt**; en caso se emita cero, se mostrará el mensaje «No hay registros..!!»; caso contrario, se leerán los registros línea por línea; la función **explode** rompe la línea y envía la información a la variable **$linea**; esta determinará que es una línea de registro completa solo cuando encuentre la combinación de funciones **chr(13).chr(10)**.

La estructura **for** permite recorrer por todas las líneas registradas en la variable **$linea**. Para poder obtener la información se deberá implementar una variable tipo arreglo, que llamaremos **$palabra**, y con la función **explode** dividiremos todas las variables para poder imprimirlas con la instrucción **$palabra[0]** para el código del cliente, **$palabra[1]** para el nombre del cliente y así sucesivamente.

```
function contador($archivo){
     $clientes = fopen($archivo,'r');
     $leer = fread($clientes,  filesize($archivo));
     $linea = explode(chr(13).chr(10),$leer);
     $n = count($linea);
     return $n;
  }
```

La función contador determina el número de registro del cliente actual, esta función ayuda en el registro de los clientes, ya que desde aquí podemos generar los números consecutivos de los registros.

Sólo fines educativos - FreeLibros

CAP.

9

Sesiones

Sólo fines educativos - FreeLibros

Sólo fines educativos - FreeLibros

9.1 INTRODUCCIÓN

En temas anteriores hemos usado variables locales en todas las aplicaciones, estas se caracterizan por obtener un valor y guárdalo en la memoria mientras la aplicación se esté ejecutando; es decir, el valor se destruye para una segunda invocación de la misma aplicación. En términos de programación, estaríamos hablando de que si necesitamos pasar información de variables de una página a otra, no podríamos usar variables locales. El envío de información pasa de una página a otra mediante la instrucción **$_POST**, pero se pierde a la tercera página, veamos el siguiente escenario:

Archivo: **registro.php**

```
<section>
     <form name="frmRegistro" method="POST" action="captura.php">
          <table border="0" width="500" cellspacing="1" cellpadding="1">
             <tr>
                <td width="120">Nombre del cliente</td>
                <td><input type="text" name="txtCliente" size="40"/></td>
             </tr>
          </table>
     </form>
</section>
```

Para capturar el valor ingresado usamos el siguiente código en un archivo llamado **captura.php**

```
<?php
    $nombre = $_POST['txtCliente'];
    echo 'El nombre del cliente es: '.$nombre;
    echo '<a href="remuestra.php">Enviar el nombre del cliente</a>';
?>
```

Ahora, si esta misma variable es usada en una tercera página, por ejemplo **remuestra.php** que contiene el siguiente código:

```
<?php
    $nombre = $_POST['txtCliente'];
    echo 'El nombre del cliente es: '.$nombre;
?>
```

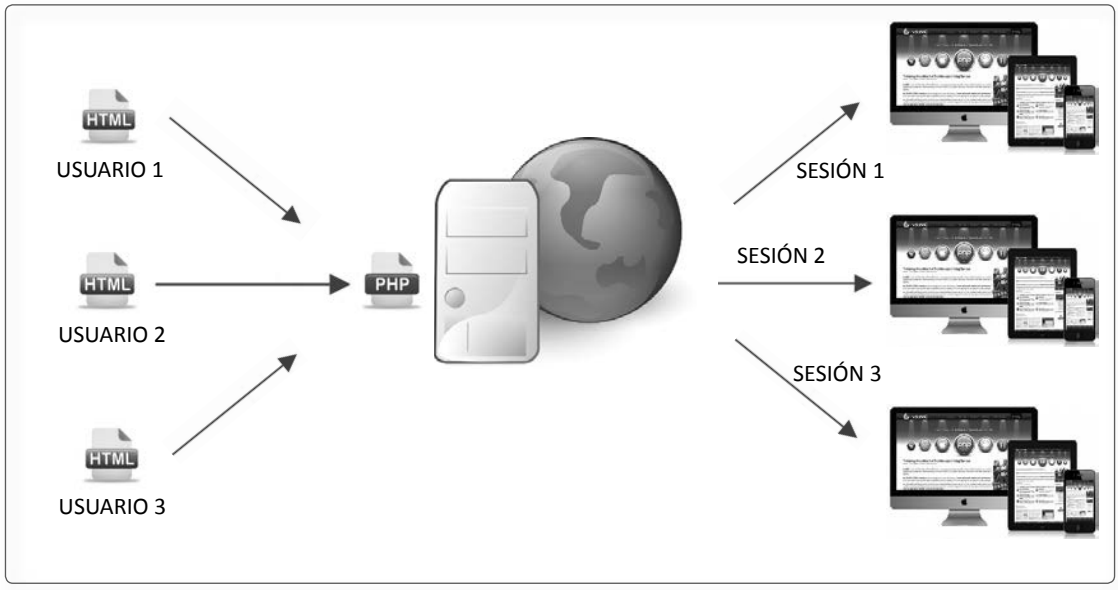
Esta emitirá un error, puesto que la variable **txtCliente** no es reconocida; ahora, se preguntará por qué no invocamos a la variable **$nombre**, esa es justamente la respuesta correcta a este caso, pero aquí nace el tema de variables globales o superglobales, como se le suele llamar a una session en PHP. Esta nos permitirá enviar información a una variable global y esta podrá ser invocada en cualquier página web que inicie su línea de código con la instrucción **session_start**.

Veremos en este capítulo el uso de las sesiones con variables simples y arreglos asociativos con casos simples, como determinar el número de visitas de una página web o implementando un carrito de compras con session de PHP.

Sólo fines educativos - FreeLibros

9.2 Definición de sesiones

Una session de PHP se caracteriza por almacenar información permanente para los distintos usuarios de nuestra web, eso quiere decir que sin necesidad de *logearse*, cada usuario que usa una session tendrá un identificador distinto. Es justamente este identificador el que permite obtener información de un determinado usuario; el código generado es aleatorio para cada usuario, por lo tanto, genera seguridad en las aplicaciones web.



En la imagen podemos observar que para cada usuario que solicita una sesión, el servidor web asigna un identificador y un nombre de session único para cada uno, generando así un nivel de acceso controlado a una aplicación.

9.3 Funciones de session

9.3.1 Función session_start()

Cumple dos funciones dentro de una aplicación PHP: la primera, consiste en iniciar una nueva session; la segunda, en reanudar la existente. Eso quiere decir que las páginas que necesiten enviar o recibir información de la sesión deben incluir la función **session_start** al inicio del código. Su formato es:

```
session_start();
```

Veamos los siguientes casos:

- Iniciar la sesión en un documento PHP puro:

```
<?php
    session_start();
?>
```

Sólo fines educativos - FreeLibros

- Iniciar la sesión en un documento HTML5 embebido con PHP:

```
<?php
    session_start();
?>
<!DOCTYPE html>
<html>
    <head>
        <meta charset="UTF-8">
    </head>
    <body>
        <header> </header>
        <section></section>
        <footer> </footer>
    </body>
</html>
```

9.3.2 Función session_id()

Permite devolver el identificador de la sesión iniciada con la función **session_start**; devolverá vacío si la sesión no fue iniciada. Su formato es:

```
session_id();
```

Veamos el siguiente caso:

- Se necesita mostrar el identificador de la sesión actual:

```
<?php
  session_start();
  echo 'El identificador de la sesión es: ' . session_id();
?>
```

El resultado es:

El identificador de la sesión es: 6k9eak1tbjl3i1auqbg3qgvvo6

Cada sesión iniciada tiene un identificador único, y esta se mantiene todo el tiempo que el usuario use la página web. Para poder generar códigos de sesión distintos podemos usar el siguiente código:

```
<?php
   session_start();

   //Capturando el identificador de la sesión actual
   $idSesionActual = session_id();

   //Regenerando el identificador
   session_regenerate_id();

   //Capturando el identificador de la nueva sesión
   $idSesionNueva = session_id();

   //Imprimir los identificadores
   echo "Identificador de la sesión anterior es:
                                   $idSesionActual<br />";
   echo "Identificador de la sesión nueva es:
                                   $idSesionNueva<br />";
?>
```

Sólo fines educativos - FreeLibros

El resultado es:

```
Identificador de la sesión anterior es: p2heos0lvrf9ua753f98habfo1
Identificador de la sesión nueva es: k5ehn3bnesc6jht04iatd8jt84
```

9.3.3 Función session_name()

Así como existe un identificador de la sesión, también existe el nombre de una sesión que se encuentra formada por un conjunto de caracteres. Su formato es:

```
session_name();
```

Veamos el siguiente caso:

- Mostrar el nombre de la sesión actual:

```
<?php
  session_start();

  //Capturando el nombre de la sesión actual
  $nombreSesion = session_name();

  //Imprimir los identificadores
  echo "Nombre de la sesión actual es: $nombreSesion<br />";
?>
```

El resultado es:

```
Nombre de la sesión actual es: PHPSESSID
```

9.3.4 Función session_unset()

Permite liberar todas las variables registradas dentro de una session de PHP. Su formato es:

```
session_unset();
```

Veamos el siguiente caso:

- Se necesita liberar todas las variables de la sesión actual:

```
<?php
   session_start();
   session_unset();
   echo 'variables descargados desde la sesión ..!!';
?>
```

9.3.5 Función session_destroy()

Permite destruir completamente un session de PHP; debe considerar que **session_destroy** no libera variables registradas en la sesión. Su formato es:

```
session_destroy();
```

Sólo fines educativos - FreeLibros

Veamos el siguiente caso:

- Se necesita destruir la sesión actual:

```
<?php
    session_start();
    session_unset();
    session_destroy();
    echo 'Fin de la sesión ..!!';
?>
```

9.4 ESCRITURA Y LECTURA DE UNA VARIABLE DE SESSION

Cuando una variable es enviada a la sesión es conocida como variable superglobal o variable automática. El formato para el envío de variables a la sesión es:

```
$_SESSION['variable'] = valor;
```

Debe tener en cuenta que para enviar información a la sesión debe iniciar la función **sesión_start**, de la misma forma si queremos recuperar información desde la sesión, usaremos el siguiente formato:

```
$variable = $_SESSION['variable'];
```

Veamos el siguiente caso:

- Se necesita enviar el nombre del cliente «Fernanda Ximena Torres Lázaro» a la sesión desde el archivo **envia.php**; luego, por medio de un enlace, se debe mostrar la información desde el archivo **recupera.php**.

envia.php

```
<?php
    session_start();
    $_SESSION['cliente']='Fernanda Ximena Torres Lázaro';
    echo 'Valor enviado a la session correctamente<br>';
    echo '<a href="recupera.php">Recuperar información</a>';
?>
```

Comentarios:

El nombre del cliente «Fernanda Ximena Torres Lázaro» es asignado a la variable de sesión «cliente»: en caso la variable no se encuentre en la sesión, la crea y asigna el valor del nombre del cliente; caso contrario, sobrescribe el valor.

recupera.php

```
<?php
    session_start();
    $nombre = $_SESSION['cliente'];
    echo 'Valor obtenido de la sesión es: '.$nombre;
    echo '<a href="envia.php">Enviar información</a>';
?>
```

Sólo fines educativos - FreeLibros

Comentarios:

La variable **$nombre** recupera la última información registrada en la variable «cliente» desde la sesión. Tome en cuenta que al inicio de ambos archivos debe encontrarse la función **session_start**.

9.5 Escritura y lectura de un arreglo unidimensional en la session

Una sesión puede recibir un bloque de valores mediante un arreglo de elementos, su tratamiento es similar al de una variable de sesión. El formato es:

```
$_SESSION['variable_arreglo'] = $arreglo;
```

Para obtener información del arreglo que se encuentra en la sesión, debemos usar el siguiente formato:

```
$arreglo = $_SESSION['variable_arreglo'];
```

Veamos el siguiente caso:

- Se necesita enviar información de los siguientes productos: Lavadora, Refrigeradora, DVD, Televisor y Radiograbadora a la sesión desde el archivo **envia.php**; luego, por medio de un enlace, se debe mostrar la información desde el archivo **recupera.php**.

envia.php

```
<?php
    session_start();
    $productos = array('Lavadora','Refrigeradora',
                       'DVD','Televisor','Radiograbadora');

    $_SESSION['misProductos']=$productos;
    echo 'Arreglo de productos enviado a la
                              session correctamente<br>';
    echo '<a href="recupera.php">Recuperar información</a>';
?>
```

Comentarios:

Se crea una variable de tipo arreglo llamado **$productos**, la cual almacena los nombres de los productos y es enviada a la session con el nombre de **misProductos**.

recupera.php

```
<?php
    session_start();
    $productos = $_SESSION['misProductos'];

    echo 'Los productos del arreglo son: ';
    foreach($productos as $producto){
        echo $producto.'<br>';
    }
    echo '<a href="envia.php">Enviar información</a>';
?>
```

Sólo fines educativos - FreeLibros

Comentarios:

Se inicia la sesión y se descargan los productos desde el arreglo **misProductos**, que se encuentra en la sesión, a la variable **$productos**; luego, para ser impresos, debemos usar la estructura **foreach**. También podríamos usar el siguiente script para imprimir:

```
<?php
    session_start();

    echo 'Los productos del arreglo son: ';
    foreach($_SESSION['misProductos'] as $producto){
        echo $producto.'<br>';
    }
    echo '<a href="envia.php">Enviar información</a>';
?>
```

9.6 ESCRITURA Y LECTURA DE UN ARREGLO ASOCIATIVO EN LA SESSION

Una sesión también puede recibir un arreglo asociativo de dos o más dimensiones, su tratamiento es similar al de una variable de sesión. El formato es:

```
$_SESSION['variable_arreglo_asociativo'] = $arreglo_asociativo;
```

Para obtener información del arreglo que se encuentra en la sesión, debemos usar el siguiente formato:

```
$arreglo_asociativo = $_SESSION['variable_arreglo_asociativo'];
```

Veamos el siguiente caso:

- Se necesita enviar información de los siguientes productos:

Descripción del producto	Precio
Lavadora	$1500.00
Refrigeradora	$2000.00
DVD	$ 200.00
Televisor	$1500.00
Radiograbadora	$3500.00

Dicha información debe ser enviada a la sesión desde el archivo **envia.php**; luego, por medio de un enlace, se debe mostrar la información desde el archivo **recupera.php**.

envia.php

```
<?php
    session_start();
    $productos = array('Lavadora'=>1500,'Refrigeradora'=>2000,
                       'DVD'=>200,'Televisor'=>1500,
                       'Radiograbadora'=>3500);

    $_SESSION['misProductos']=$productos;

    echo 'Arreglo asociativo de productos enviado a
                                la session correctamente<br>';
    echo '<a href="recupera.php">Recuperar información</a>';
?>
```

Sólo fines educativos - FreeLibros

Comentarios:

Se inicia la sesión, luego se crea el arreglo asociativo **$productos** con el nombre de los productos y sus respectivos precios; seguidamente dicha información es enviada a la session por medio de la instrucción **$_SESSION['misProductos']**. El nombre **misProductos** será la variable de sesión.

recupera.php

```
<?php
    session_start();
    echo 'Los productos del arreglo son: ';
    foreach( $_SESSION['misProductos'] as $producto=>$precio){
        echo $producto. ' '.$precio.'<br>';
    }
    echo '<a href="envia.php">Enviar información</a>';
?>
```

Comentarios:

Se restablece la sesión con **session_start** y se obtienen todos los valores, desde arreglo mediante la estructura repetitiva **foreach**, el cual envía la información del arreglo hacia las variables **$producto** y **$precio**, que luego son impresos dentro del mismo ciclo de repeticiones.

9.7 Casos desarrollados

Para todos los casos desarrollados será conveniente usar los siguientes estilos, llámelos estilo.css:

```
/* Estilo para el fondo del documento */
body{
    font-family: tahoma;
    font-size: 14px;
}

/* Estilo que permite centrar un texto*/
#centrado{
    text-align: center;
}

/* Estilo que permite centrar horizontalmente una tabla */
table {
    margin: auto;
}

/* Estilo para las imagenes */
img{
    margin: auto;
    display: block;
}

/* Estilo para dar un contorno a la celda de una tabla */
td {
    border: solid 1px #006699;
    border-top: #006699;
    border-right: #006699;
    border-left: #006699;
    border-bottom: #006699;
```

Sólo fines educativos - FreeLibros

```
}
/* Estilo para rellenar de color las celdas th de una tabla */
table th {
   background-color: #b9c9fe;
}

/* Estilo para resaltar un texto con tamaño 22px */
#resaltado{
    font-family: 'trebuchet MS' ;
    font-size: 22px ;
}

/* Estilo para alinear un texto a la derecha*/
#derecha{
    text-align: right;
}

/* Estilo para formatear textos de impresión de dinero*/
#totales{
    font-family: 'trebuchet MS' ;
    font-size: 22px ;
    text-align: right;
}
```

- **Caso desarrollado 1: Verificación de la session**

Implemente una aplicación web con PHP que permita registrar y mostrar el nombre de un libro perteneciente a una biblioteca, este será enviado y devuelto desde la misma sesión.

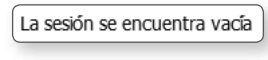
Debemos tener en cuenta:

Se debe comprobar que la variable se encuentre en la session; en caso sea así, emitir el mensaje «La sesión está llena y contiene». Caso contrario, mostrar el mensaje «La sesión se encuentra vacía».

Archivo: **verifica.php**

```
<?php
    //Iniciando la sesión
    session_start();

    //Comprobando si la variable libro tiene valor en la sesión
    if (isset($_SESSION['libro'])) {
     $unLibro=$_SESSION['libro'];
        echo 'La sesión esta llena y contiene > '.$unLibro;
    }else{
        echo 'La sesión se encuentra vacía';
        $_SESSION['libro'] = 'Tradiciones Peruanas';
    }
?>
```

Al iniciar, la aplicación mostrará la siguiente pantalla:



Al presionar F5 sobre la web se mostrará la siguiente imagen:

Sólo fines educativos - FreeLibros

Comentarios:

```
session_start();
```

Iniciando la sesión, recuerde que esta instrucción debe ser colocada al inicio de cualquier código, inclusive del código HTML.

```
if (isset($_SESSION['libro'])) {
      $unLibro=$_SESSION['libro'];
        echo 'La sesión esta llena y contiene > '.$unLibro;
    }else{
        echo 'La sesión se encuentra vacía';
        $_SESSION['libro'] = 'Tradiciones Peruanas';
    }
```

La función **isset** comprueba que la variable libro tenga algún valor en la sesión; si es verdadera la condición, entonces la variable **$unLibro** obtendrá el valor desde la sesión con la instrucción **$_SESSION['libro']**. Ahora, si en la sesión no se encuentra la variable **libro**, se creará con el título «Tradiciones Peruanas». Finalmente, cuando se inicie la aplicación se mostrará el mensaje «La sesión se encuentra vacía»; al presionar por segunda vez, mostrar el contenido de la variable **libro** desde la sesión.

- **Caso desarrollado 2: Uso de colores desde la session**

Implemente una aplicación en PHP que permita seleccionar un color desde un cuadro combinado y, dependiendo del color seleccionado, que lo muestre en una celda posterior, tal como se muestra en la siguiente imagen:

Tener en cuenta:

- Debe guardar los siguientes colores y sus códigos en un arreglo asociativo, estos serán enviados a la sesión:

Nombre del color	Código
Azul	#0000FF
Verde	#00FF00
Rojo	#FF0000
Negro	#000000

- El cambio de color en el cuadro debe darse al seleccionar un color desde el cuadro combinado; para lo cual debe asignarle la propiedad **onchange="this.form.submit()"** al cuadro combinado.

Sólo fines educativos - FreeLibros

Archivo: **colores.php**

```
<!DOCTYPE html>
<html>
<head>
  <meta charset="UTF-8">
  <title></title>
  <link href="estilo.css" rel="stylesheet">
</head>
<body>
  <header>
   <h2 id="centrado">Control de colores</h2>
   <h4 id="centrado">Usando session</h4>
  </header>
  <section>
    <form name="frmColores" method="POST" action="colores.php">
      <table width="400" height="120"
             cellspacing="10" cellpadding="0">
        <tr>
          <td>
            <table width="200" cellspacing="0" cellpadding="0">
          <tr>
          <td>Seleccione color de fondo</td>
          </tr>
          <tr>
            <td><select name="selFondo" onchange="this.form.submit()">
                        <option>(Seleccione)</option>
                        <option>Azul</option>
                        <option>Verde</option>
                        <option>Rojo</option>
                        <option>Negro</option>
                </select>
            </td>
          </tr>
       </table>
          </td>
            <?php
              error_reporting(0);
              session_start();
              $colores=array(
                           'Azul'=>'#0000FF',
                           'Verde'=>'#00FF00',
                           'Rojo'=>'#FF0000',
                           'Negro'=>'#000000');

              $color=$_POST['selFondo'];
              $_SESSION['fondo']=$colores[$color];
              $fondo = $_SESSION['fondo'] ;
             ?>
            <td bgcolor="<?php echo $fondo; ?>">
                        El color seleccionado es: <?php echo $color; ?>
            </td>
       </tr>
      </table>
     </form>
   </section>
   <footer>
        <h5 id="centrado">Todos los derechos reservados @2015
                Diseñado por M@nuel Torres</h5>
   </footer>
</body>
</html>
```

Sólo fines educativos - FreeLibros

Comentarios:

```
<select name="selFondo" onchange="this.form.submit()">
                <option>(Seleccione)</option>
                <option>Azul</option>
                <option>Verde</option>
                <option>Rojo</option>
                <option>Negro</option>
</select>
```

Definición del control **cuadro combinado**, que permitirá mostrar los colores que el usuario seleccionará para poder dar actividad **submit** al control; se habilita la propiedad **onchange** con la opción **this.form.submit**. Con esta instrucción el cuadro combinado ejecutará **submit** al momento de seleccionar un color.

```
<?php
          error_reporting(0);
          session_start();
          $colores=array(
                    'Azul'=>'#0000FF',
                    'Verde'=>'#00FF00',
                    'Rojo'=>'#FF0000',
                    'Negro'=>'#000000');

          $color=$_POST['selFondo'];
          $_SESSION['fondo']=$colores[$color];
          $fondo = $_SESSION['fondo'] ;
?>
```

Empezaremos explicando que **error_reporting** permite ocultar algunos errores de cuidado que ocasionará la aplicación; se inicia la sesión y se crea el arreglo **$colores**, y se le asignan los colores con sus respectivos códigos.

$color = $_POST['selFondo']: Captura el color seleccionado desde el cuadro combinado.

$_SESSION['fondo']=$colores[$color]: Se envía el código de color encontrado en el arreglo según el color seleccionado por el usuario, y es enviado a la sesión con el nombre **fondo**.

$fondo = $_SESSION['fondo']: Luego se obtiene el código de color desde la sesión y es enviado a la variable **fondo**, esto servirá para asignarle el color a la celda.

```
<td bgcolor="<?php echo $fondo; ?>">
       El color seleccionado es: <?php echo $color; ?>
</td>
```

La variable **$fondo** obtenida desde la sesión contiene el código del color que será asignado a la celda <TD>, mediante su propiedad **bgcolor**.

Sólo fines educativos - FreeLibros

- **Caso desarrollado 3: Encuesta de inseguridad**

Implemente una aplicación en PHP que permita registrar una encuesta de inseguridad realizada a los ciudadanos del distrito de Lima, para lo cual se realiza las siguientes preguntas:

¿Cómo se siente a nivel de seguridad en la ciudad de Lima?

¿Qué tipo de delito cree usted que es el más común en su zona de residencia?

¿Generalmente qué día cree usted que ocurren más incidencias?

¿Cuáles cree usted que deben ser las medidas que las autoridades deben aplicar para combatir estas incidencias?

¿Usted conoce las medidas que implementa la policía del Perú para disminuir la delincuencia?

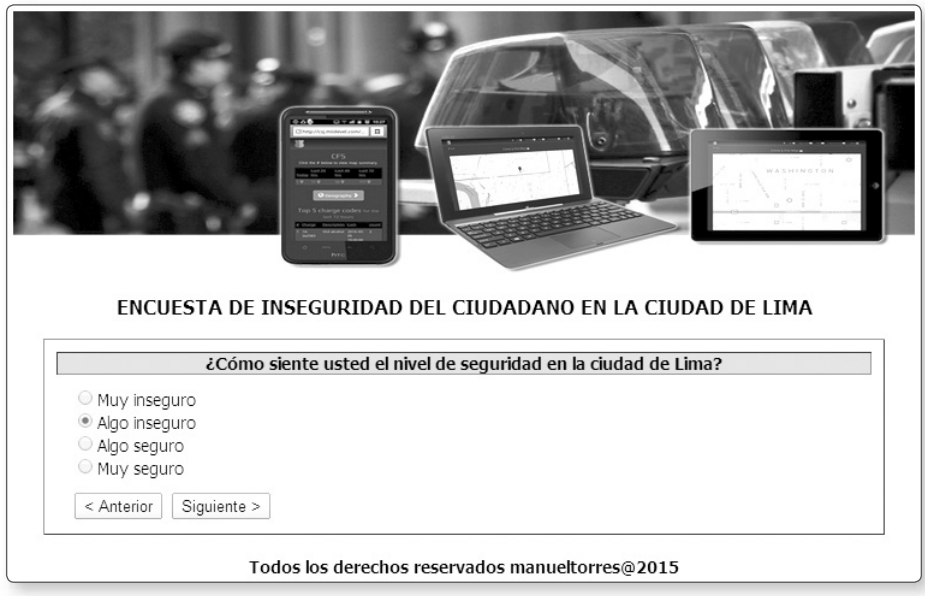
Debe tener en cuenta los siguientes aspectos:

- Al iniciar la aplicación, deberá solicitar los datos del ciudadano, como su nombre completo y su número de DNI; este último solo deberá permitir el ingreso de 8 dígitos. La interfaz debe ser como se muestra a continuación:

index.php

- Al iniciar la encuesta se mostrará la interfaz con la primera pregunta, la cual mostrará las alternativas que el usuario puede seleccionar, además de los botones **Anterior** (devuelve a la página inicial) y **Siguiente** (envía a la segunda pregunta). La interfaz se muestra a continuación:

Sólo fines educativos - FreeLibros

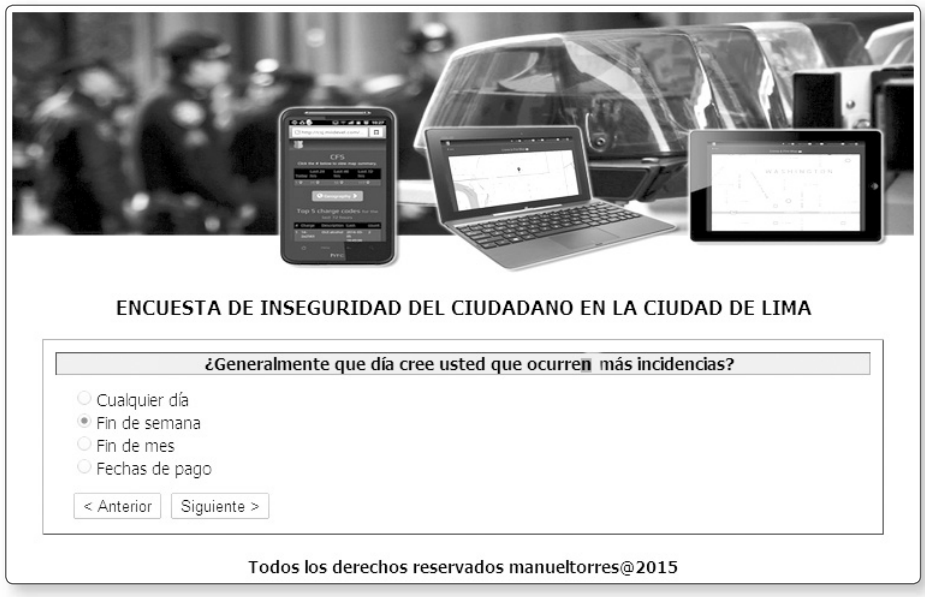


pregunta1.php

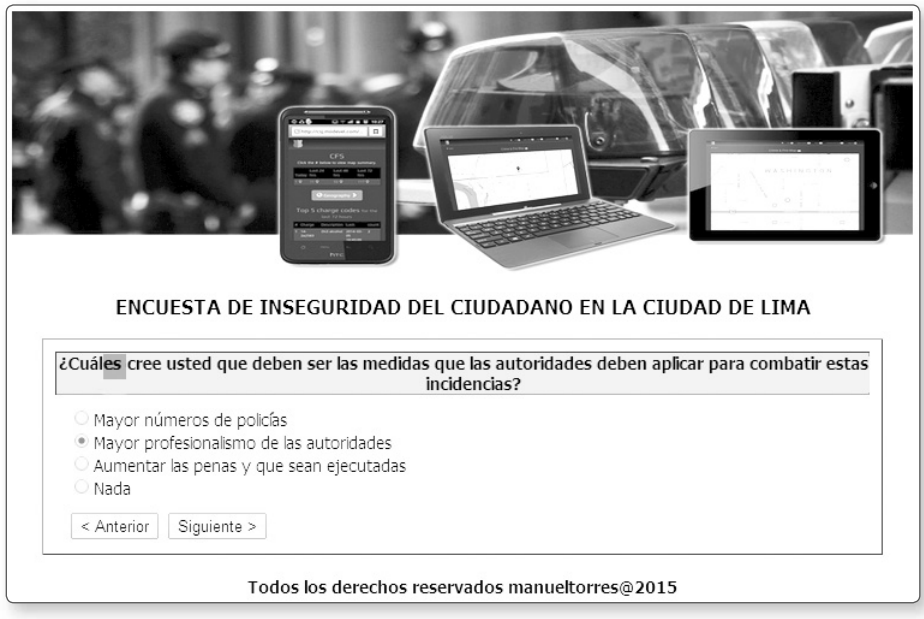
- Una vez efectuada la primera pregunta, se procederá de la misma manera para la pregunta dos, tres, cuatro y cinco; cada una con botones de **Anterior** y **Siguiente** respectivamente. Las interfaces se muestran a continuación:

pregunta02.php

Sólo fines educativos - FreeLibros



pregunta03.php



pregunta04.php

Sólo fines educativos - FreeLibros

pregunta05.php

- Al completar las cinco preguntas le aparecerá un «INFORME DE ENCUESTA», en la cual se mostrarán todas las opciones seleccionadas por el ciudadano. En esta ventana se presentará el botón **Anterior,** que permitirá retornar a la pregunta cinco; mientras que **Volver a encuestar** le devolverá a la página **index.php**. La interfaz del informe es como se muestra a continuación:

Sólo fines educativos - FreeLibros

Archivo: **index.php**

```
<?php
    session_start();
?>
<!DOCTYPE html>
<html>
    <head>
        <meta charset="UTF-8">
        <title></title>
        <link href="estilo.css" rel="stylesheet">
    </head>
    <body>
        <header>
            <?php include 'encabezado.php'; ?>
        </header>
        <section>
            <form method="POST" action="pregunta1.php">
                <table border="1" width="700"
                       cellspacing="10" cellpadding="0">
                <tr>
                    <td>Nombres y apellidos</td>
                    <td><input type="text" name="txtNombres"
                               size="50" /></td>
                </tr>
                <tr>
                    <td>Numero DNI</td>
                    <td><input type="text" name="txtDNI"
                               size="30" maxlength="8" /></td>
                </tr>
                <tr>
                    <td>Empezar con la encuesta >>> </td>
                    <td><input type="submit" value="Encuesta" /></td>
                </tr>
                </table>
            </form>
        </section>
        <footer>
            <?php include 'pie.php';?>
        </footer>
    </body>
</html>
```

Comentarios:

Tenga en cuenta la propiedad **action** del formulario, pues se enviarán las respuestas a la página **pregunta1.php**, y así debe ser implementada. También debe asignar nombres a los controles, como **txtNombres** para el nombre del cliente; y **txtDNI** para el ingreso del DNI con su propiedad **maxlength** para controlar los 8 caracteres.

encabezado.php

```
<!DOCTYPE html>
<html>
    <head>
        <meta charset="UTF-8">
        <title></title>
        <link href="estilo.css" rel="stylesheet">
    </head>
    <body>
```

Sólo fines educativos - FreeLibros

```
        <img src="banner_seguridad.png" width="750" height="220"/>
        <h3 id="centrado">ENCUESTA DE INSEGURIDAD DEL CIUDADANO
                          EN LA CIUDAD DE LIMA
        </h3>
    </body>
</html>
```

Comentarios:

La página encabezado contiene información que pertenece a todos los encabezados de las páginas que componen el proyecto; este será referenciado por todas las páginas, con la instrucción **include** de PHP. Considere que la imagen *banner* lo puede descargar desde la galería de imágenes google, y que el ancho debe ser el mismo que se configura en la tabla para que tenga una distribución adecuada.

pie.php

```
<!DOCTYPE html>
<html>
    <head>
        <meta charset="UTF-8">
        <title></title>
        <link href="estilo.css" rel="stylesheet">
    </head>
    <body>
        <h4 id="centrado">Todos los derechos reservados
                          manueltorres@2015</h4>
    </body>
</html>
```

Comentarios:

De la misma forma que el archivo encabezado, el pie se insertará en todas las páginas del proyecto, solo deberá indicar el nombre del autor y será referenciado por la instrucción **include** de PHP.

Archivo: **pregunta1.php**

```
<?php
    session_start();
    $_SESSION['nombres']=$_POST['txtNombres'];
    $_SESSION['dni']=$_POST['txtDNI'];
?>
<!DOCTYPE html>
<html>
    <head>
        <meta charset="UTF-8">
        <title></title>
        <link href="estilo.css" rel="stylesheet">
    </head>
    <body>
        <header>
            <?php include 'encabezado.php'; ?>
        </header>
        <section>
            <form method="POST" action="pregunta1.php">
                <table border="1" width="700"
```

Sólo fines educativos - FreeLibros

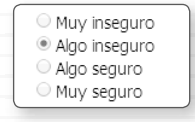
```
                    cellspacing="10" cellpadding="0">
                <tr>
                    <th colspan="2">¿Cómo siente usted el nivel
                                    de seguridad en la ciudad de Lima?
                    </th>
                </tr>
                <tr>
                    <td></td>
                    <td>
                        <input type="radio" name="preg1"
                            value="Muy inseguro" />Muy inseguro<br/>
                        <input type="radio" name="preg1"
                            value="Algo inseguro" />Algo inseguro<br/>
                        <input type="radio" name="preg1"
                            value="Algo seguro" />Algo seguro<br/>
                        <input type="radio" name="preg1"
                            value="Muy seguro" />Muy seguro
                    </td>
                </tr>
                <tr>
                    <td></td>
                    <td>
                        <input type="submit" value="< Anterior"
                             onclick="this.form.action='index.php'" />
                        <input type="submit" value="Siguiente >"
                             onclick="this.form.action='pregunta2.php'"/>
                    </td>
                </tr>
                </table>
            </form>
        </section>
        <footer>
            <?php include 'pie.php';?>
        </footer>
    </body>
</html>
```

Comentarios:

```
<?php
    session_start();
    $_SESSION['nombres']=$_POST['txtNombres'];
    $_SESSION['dni']=$_POST['txtDNI'];
?>
```

Se inicia la sesión para la pregunta uno con **session_start**, se obtiene los nombres y dni desde la página **index.php**; estos valores serán enviados a la session como «nombres» y «dni», respectivamente.

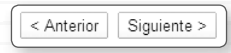
```
<td>
    <input type="radio" name="preg1" value="Muy inseguro" />Muy inseguro<br/>
    <input type="radio" name="preg1" value="Algo inseguro" />Algo inseguro<br/>
    <input type="radio" name="preg1" value="Algo seguro" />Algo seguro<br/>
    <input type="radio" name="preg1" value="Muy seguro" />Muy seguro
</td>
```



Sólo fines educativos - FreeLibros

El código implementa los botones de opción que el usuario podrá seleccionar al contestar una pregunta de la encuesta, para lo cual se debe dar el mismo nombre para todos los controles. Tal como se muestra en el código donde todas las opciones tienen el nombre de **preg1**, tiene que ser de esta forma para que al seleccionar una opción se desmarquen las otras opciones; la propiedad **value** es la más importante pues es desde aquí de donde obtenemos que opción que seleccionó el usuario.

```
<td>
 <input type="submit" value="< Anterior" onclick="this.form.action='index.php'" />
 <input type="submit" value="Siguiente >" onclick="this.form.action='pregunta2.php'"/>
</td>
```



Una de las condiciones de la aplicación recalca que debemos colocar dos botones de tipo **Action**; es decir, cada uno enviará a páginas distintas, para lo cual se deberá asignar la propiedad **onclick="this.form.action='index.php'"**, el cual permite retornar a la página **index.php**; de la misma forma para la página **pregunta2.php**.

Archivo: **pregunta2.php**

```
<?php
    session_start();
    $_SESSION['Pregunta1']=$_POST['preg1'];
?>
<!DOCTYPE html>
<html>
    <head>
        <meta charset="UTF-8">
        <title></title>
        <link href="estilo.css" rel="stylesheet">
    </head>
    <body>
        <header>
            <?php include 'encabezado.php'; ?>
        </header>
        <section>
            <form method="POST" action="pregunta2.php">
                <table border="1" width="700"
                       cellspacing="10" cellpadding="0">
                <tr>
                    <th colspan="2">¿Qué tipo de delito cree usted
                                    que es el más
                                    común en su zona de residencia?</th>
                </tr>
                <tr>
                    <td></td>
                    <td>
                        <input type="radio" name="preg2"
                               value="Hurto" />Hurto<br/>
                        <input type="radio" name="preg2"
                               value="Robo por descuido" />
                               Robo por descuido<br/>
                        <input type="radio" name="preg2"
                               value="Robo de vehículo" />
                               robo de vehículo<br/>
                        <input type="radio" name="preg2"
                               value="Intento de hurto" />
                               Intento de hurto<br/>
```

Sólo fines educativos - FreeLibros

```
                        <input type="radio" name="preg2"
                               value="Estafa" />Estafa<br/>
                        <input type="radio" name="preg2"
                               value="Asesinato" />Asesinato
                    </td>
                </tr>
                <tr>
                    <td></td>
                    <td>
                        <input type="submit"
                             value="< Anterior"
                             onclick="this.form.action='pregunta1.php'"/>
                        <input type="submit" value="Siguiente >"
                             onclick="this.form.action='pregunta3.php'"/>
                    </td>
                </tr>
                </table>
            </form>
        </section>
        <footer>
            <?php include 'pie.php';?>
        </footer>
    </body>
</html>
```

Comentarios:

```
<?php
 session_start();
 $_SESSION['Pregunta1']=$_POST['preg1'];
?>
```

Se inicia la sesión con la sentencia **session_start** y se envía a dicha sesión la respuesta seleccionada por el usuario.

Archivo: **pregunta3.php**

```
<?php
    session_start();
    $_SESSION['Pregunta2']=$_POST['preg2'];
?>
<!DOCTYPE html>
<html>
  <head>
     <meta charset="UTF-8">
     <title></title>
     <link href="estilo.css" rel="stylesheet">
  </head>
  <body>
     <header>
        <?php include 'encabezado.php'; ?>
     </header>
     <section>
        <form method="POST" action="pregunta3.php">
           <table border="1" width="700" cellspacing="10" cellpadding="0">
             <tr>
```

Sólo fines educativos - FreeLibros

```
                <th colspan="2">¿Generalmente que día cree usted que
                                ocurre más incidencias?</th>
              </tr>
              <tr>
                 <td></td>
                 <td>
                   <input type="radio" name="preg3"
                          value="Cualquier día" />Cualquier día<br/>
                   <input type="radio" name="preg3"
                          value="Fin de semana" />Fin de semana<br/>
                   <input type="radio" name="preg3"
                          value="Fin de mes" />Fin de mes<br/>
                   <input type="radio" name="preg3"
                          value="Fechas de pago" />Fechas de pago
                 </td>
              </tr>
              <tr>
                 <td></td>
                 <td>
                    <input type="submit"
                           value="< Anterior"
                           onclick = "this.form.action = 'pregunta2.php'" />
                    <input type="submit"
                           value="Siguiente >"
                           onclick = "this.form.action = 'pregunta4.php'" />
                 </td>
              </tr>
           </table>
        </form>
     </section>
         <footer>
             <?php include 'pie.php';?>
         </footer>
     </body>
</html>
```

Archivo: **pregunta4.php**

```
<?php
    session_start();
    $_SESSION['Pregunta3']=$_POST['preg3'];
?>
<!DOCTYPE html>
<html>
  <head>
     <meta charset="UTF-8">
     <title></title>
     <link href="estilo.css" rel="stylesheet">
  </head>
  <body>
     <header>
         <?php include 'encabezado.php'; ?>
     </header>
     <section>
       <form method="POST" action="pregunta4.php">
          <table border="1" width="700" cellspacing="10" cellpadding="0">
            <tr>
               <th colspan="2">¿Cuál cree usted que deben ser las medidas
                               que las autoridades deben aplicar para
                               combatir estas incidencias?</th>
```

Sólo fines educativos - FreeLibros

```
            </tr>
            <tr>
               <td></td>
               <td>
                 <input type="radio" name="preg4"
                        value="Mayor números de policías" />
                        Mayor números de policías<br/>
                 <input type="radio" name="preg4"
                        value="Mayor profesionalismo de las autoridades" />
                        Mayor profesionalismo de las autoridades<br/>
                 <input type="radio" name="preg4"
                        value="Aumentar las penas y que sean ejecutadas" />
                        Aumentar las penas y que sean ejecutadas<br/>
                 <input type="radio" name="preg4" value="Nada" />Nada
               </td>
            </tr>
            <tr>
               <td></td>
               <td>
                  <input type="submit" value="< Anterior"
                         onclick = "this.form.action = 'pregunta3.php'" />
                  <input type="submit" value="Siguiente >"
                         onclick = "this.form.action = 'pregunta5.php'" />
               </td>
            </tr>
         </table>
      </form>
   </section>
   <footer>
       <?php include 'pie.php';?>
   </footer>
 </body>
</html>
```

Archivo: **pregunta5.php**

```
<?php
    session_start();
    $_SESSION['Pregunta4']=$_POST['preg4'];
?>
<!DOCTYPE html>
<html>
  <head>
    <meta charset="UTF-8">
    <title></title>
    <link href="estilo.css" rel="stylesheet">
  </head>
  <body>
    <header>
       <?php include 'encabezado.php'; ?>
    </header>
    <section>
      <form method="POST" action="pregunta5.php">
         <table border="1" width="700" cellspacing="10" cellpadding="0">
           <tr>
             <th colspan="2">¿Usted conoce las medidas que implementa la
                        policía del Perú para disminuir la delincuencia?</th>
           </tr>
           <tr>
             <td></td>
```

Sólo fines educativos - FreeLibros

```
            <td>
              <input type="radio" name="preg5" value="Si" />Si<br/>
              <input type="radio" name="preg5" value="No" />No<br/>
            </td>
          </tr>
          <tr>
            <td></td>
            <td>
              <input type="submit" value="< Anterior"
                     onclick = "this.form.action = 'pregunta4.php'" />
              <input type="submit" value="Siguiente >"
                     onclick = "this.form.action = 'resumen.php'" />
            </td>
          </tr>
        </table>
      </form>
   </section>
   <footer>
      <?php include 'pie.php';?>
   </footer>
 </body>
</html>
```

resumen.php

```
<?php
    session_start();
    $_SESSION['Pregunta5']=$_POST['preg5'];
?>
<!DOCTYPE html>
<html>
  <head>
    <meta charset="UTF-8">
    <title></title>
    <link href="estilo.css" rel="stylesheet">
  </head>
  <body>
    <header>
       <?php include 'encabezado.php'; ?>
       <h3>INFORME DE ENCUESTA</h3>
    </header>
    <?php
      $nombres = $_SESSION['nombres'];
      $dni = $_SESSION['dni'];
      $pregunta1 = $_SESSION['Pregunta1'];
      $pregunta2= $_SESSION['Pregunta2'];
      $pregunta3 = $_SESSION['Pregunta3'];
      $pregunta4 = $_SESSION['Pregunta4'];
      $pregunta5 = $_SESSION['Pregunta5'];
    ?>
   <section>
     <form method="POST" action="resumen.php">
       <table border="1" width="700" cellspacing="10" cellpadding="0">
          <tr>
            <td>Nombre del ciudadano</td>
            <td><?php echo $nombres; ?></td>
          </tr>
          <tr>
            <td>DNI</td>
            <td><?php echo $dni; ?></td>
```

Sólo fines educativos - FreeLibros

```
            </tr>
            <tr>
              <td>¿Cómo se siente a nivel de seguridad en la ciudad de Lima?
              </td>
              <td><?php echo $pregunta1; ?></td>
            </tr>
            <tr>
              <td>¿Qué tipo de delito cree usted que es el más
                  común en su zona de residencia?</td>
              <td><?php echo $pregunta2; ?></td>
            </tr>
            <tr>
              <td>¿Generalmente que día cree usted que ocurre
                  más incidencias?</td>
              <td><?php echo $pregunta3; ?></td>
            </tr>
            <tr>
              <td>¿Cuál que usted que deben ser las medidas que las
                   autoridades deben aplicar para combatir estas
                   incidencias?</td>
              <td><?php echo $pregunta4; ?></td>
            </tr>
            <tr>
              <td>¿Usted conoce las medidas que implementa la policía
                  del Perú para disminuir la delincuencia?</td>
              <td><?php echo $pregunta5; ?></td>
            </tr>
            <tr>
              <td>
                  <input type="submit" value="< Anterior"
                         onclick = "this.form.action = 'pregunta5.php'" />
                  <input type="submit" value="Volver a encuestar"
                         onclick = "this.form.action = 'index.php'" />
                      </td>
                  </tr>
                  </table>

              </form>
          </section>
          <footer>
              <?php include 'pie.php';?>
          </footer>
      </body>
  </html>
```

Comentarios:

```
session_start();
$_SESSION['Pregunta5']=$_POST['preg5'];
```

Iniciamos la sesión con la sentencia **session_start()** y finalmente se captura la respuesta seleccionada por el usuario para la pregunta número cinco.

Sólo fines educativos - FreeLibros

```
$nombres = $_SESSION['nombres'];
$dni = $_SESSION['dni'];
$pregunta1 = $_SESSION['Pregunta1'];
$pregunta2= $_SESSION['Pregunta2'];
$pregunta3 = $_SESSION['Pregunta3'];
$pregunta4 = $_SESSION['Pregunta4'];
$pregunta5 = $_SESSION['Pregunta5'];
```

La sesión cuenta con variables registradas en cada uno de las preguntas contestadas por el usuario; de la página de resumen se deberá obtener dichos valores con la sentencia **$_SESSION['Variable']**.

```
<tr>
   <td>Nombre del ciudadano</td>
   <td><?php echo $nombres; ?></td>
 </tr>
```

Se imprime el nombre del ciudadano, el cual fue obtenido desde la session actual; de la misma forma se realizará para la impresión de las respuestas seleccionadas en la encuesta.

- **Caso desarrollado 4: Login de usuario**

Implemente una aplicación web con PHP que permita tener el control de los clientes, para lo cual, al iniciar, deberá mostrar una ventana de acceso solicitando el nombre del usuario y su clave. Si todo es correcto, se mostrará una ventana de control de cliente en la cual pueda registrar al nuevo cliente o listar a los que ya se encuentran registrados.

Tener en cuenta:

- Implementar toda la aplicación en una carpeta llamada **Login**.
- Deberá definir los siguientes archivos:
  - **index.php**: Presenta la interfaz de validación de usuario y su clave, hay que tener en cuenta que la clave solo pueden ser cuatro dígitos.

Sólo fines educativos - FreeLibros

En caso el usuario o la clave sean incorrectos, se deberá mostrar el mensaje «Usuario incorrecto» tal como se muestra en la siguiente imagen:

- **principal.php:** Contiene la página principal del proyecto e incluye los archivos **encabezado.php** y **pie.php**; debemos tener en cuenta que esta página mostrará el nombre del usuario *logeado* y un enlace al cierre de sesión, el cual cerrará la sesión para finalmente direccionarlo a la página **index.php**:

Recuerde que el objetivo del caso se centra en el *logeo* del usuario, por lo tanto los enlaces originales de la página serán de la siguiente manera **<a href="#">**.

Sólo fines educativos - FreeLibros

Las imágenes usadas en todo el proyecto tienen los siguientes nombres:

icono_cliente.png

icono_lista.png

icono_salida.png

Usuario.png

Archivo: **index.php**

```
<!DOCTYPE html>
<html>
    <head>
        <meta charset="UTF-8">
        <title></title>
        <link href="estilo_index.css" rel="stylesheet">
    </head>
    <body>
        <section>
        <?php
          error_reporting(0);
          session_start();

          if (!isset($_SESSION['admin'])){
        ?>
            <form name="frmLogin" method="POST" action="login.php">
            <table border="3"  cellspacing="0" cellpadding="5">
            <tr>
                <td colspan="3">
                    <p id="titulo">Acesso<p>
                </td>
            </tr>
            <tr>
                <td id="derecha" width="150">Usuario</td>
                <td><input type="text" name="txtUsuario"
                           value="" /></td>
                <td rowspan="4">
                    <img src="usuario.png" width="200" height="200" />
                </td>
            </tr>
            <tr>
                <td id="derecha">Clave</td>
                <td>
                    <input type="password" name="txtClave" maxlength="4" />
                </td>
            </tr>
            <tr>
                <td></td>
                <td>
```

Sólo fines educativos - FreeLibros

```
                    <input type="checkbox" name="" value="ON" />
                           Recordar la clave
                </td>
            </tr>
            <tr>
                <td></td>
                <td>
                    <input id="boton" type="submit"
                           name="btnLogin" value=" LOGIN " />
                </td>
            </tr>
            <tr>
                <td></td>
                <td>
                 <?php
                    }else{
                        header('location:principal.php');
                    }
                    if (isset($_SESSION['error'])){
                        echo $_SESSION['error'];
                        unset($_SESSION['error']);
                    }
                 ?>
                </td>
            </tr>
            </table>
          </form>
        </section>
    </body>
</html>
```

Archivo: **login.php**

```
<?php
    session_start();

    if ($_POST['txtUsuario']=='pmtorres' and $_POST['txtClave']=='php')
        $_SESSION['admin']=$_POST['txtUsuario'];
    else
        $_SESSION['error']='Usuario incorrecto';

    header('location:index.php');
?>
```

Archivo: **cerrar.php**

```
<?php
    session_start();
    if (isset($_SESSION['admin'])){
        unset($_SESSION['admin']);
    }
    header('location:index.php');
?>
```

Sólo fines educativos - FreeLibros

Archivo: **principal.php**

```
<!DOCTYPE html>
<html>
    <head>
        <meta charset="UTF-8">
        <title>CONTROL DE CLIENTES</title>
        <link href="estilo_principal.css" rel="stylesheet">
    </head>
    <body>
        <header>
            <?php
                session_start();
                require 'encabezado.php';
                echo "<p id='centrado'>";
                echo 'Bienvenido > '.$_SESSION['admin'].'<br>';
                echo "|<a href='cerrar.php'> Cerrar Sesion </a>|";
                echo '</p>';
            ?>
        </header>
        <section>
            <table border="1" width="550" cellspacing="5">
            <tr>
                <td><a href="#">
                          <img src="icono_cliente.png"
                               width="60" height="60" /></a>
                </td>
                <td><a href="#">Registro de clientes</a></td>
            </tr>
            <tr>
                <td><a href="#">
                          <img src="icono_lista.png"
                               width="60" height="60" /></a>
                </td>
                <td><a href="#">Listado de Clientes</a></td>
            </tr>
            <tr>
                <td><a href="javascript:close()">
                       <img src="icono_salida.png"
                            width="60" height="60" /></a>
                </td>
                <td><a href="javascript:close()">Salir</a></td>
            </tr>
            </table>
        </section>
        <footer>
            <?php require 'pie.php'; ?>
        </footer>
    </body>
</html>
```

Archivo: **encabezado.php**

```
<!DOCTYPE html>
<html>
    <head>
        <meta charset="UTF-8">
        <title></title>
    </head>
    <body>
        <h2 id="centrado">CONTROL DE CLIENTES</h2>
        <img src="banner_clientes.jpg" width="550" height="120" />
    </body>
</html>
```

Sólo fines educativos - FreeLibros

Archivo: **pie.php**

```
<!DOCTYPE html>
<html>
    <head>
        <meta charset="UTF-8">
        <title></title>
    </head>
    <body>
        <p id="centrado">
            <a href="principal.php">Principal</a> |
            <a href="#">Registro del nuevo clientes</a> |
            <a href="#">Listado de clientes</a> |
            <a href="javascript:close()">Salir</a>
        </p>
        <h5 id="centrado">Todos los derechos reservados @2015
                          Diseñado por M@nuel Torres</h5>
    </body>
</html>
```

Comentarios:

```
<?php
  error_reporting(0);
  session_start();

  if (!isset($_SESSION['admin'])){
?>
```

Omitimos los mensajes de advertencia de PHP, luego iniciamos la sesión con la sentencia **session_start()**. Condicionamos la muestra del formulario **login** solo si la variable **admin** no tiene valor alguno; tenga en cuenta que la función **$_SESSION** obtiene los valores registrados en la sesión, **admin** representa la variable de sesión.

```
<?php
  }else{
    header('location:principal.php');
  }
  if (isset($_SESSION['error'])){
  echo $_SESSION['error'];
    unset($_SESSION['error']);
  }
?>
```

Si la variable **admin** se encuentra llena de valores; entonces lo direccionamos al archivo **principal.php** para poder evaluar si los valores recibidos son los correctos para el acceso del usuario.

También verificamos si la variable error, almacenada en la session tiene algún valor (**isset($_SESSION['error'])**); si es así, se mostrará en una impresión dicho mensaje y se descargará la variable **error** de la sesión **active** con la sentencia unset($_SESSION['error']).

Sólo fines educativos - FreeLibros

```
<?php
  session_start();

  if ($_POST['txtUsuario']=='pmtorres' and $_POST['txtClave']=='php')
      $_SESSION['admin']=$_POST['txtUsuario'];
  else
      $_SESSION['error']='Usuario incorrecto';

  header('location:index.php');
?>
```

Iniciamos la sesión, luego comprobamos que los valores registrados por el usuario sean correctos; para este caso, el usuario habilitado es **pmtorres** con su clave php; estos son comparados con los valores que el usuario registró en el formulario. Si son correctos, el nombre del usuario es enviado a la sesión. Caso contrario, llena la variable **error** con el mensaje «Usuario incorrecto..!!», que a su vez es impreso en la página **index.php**, por ese motive es que direccionamos a la página **index.php** en la última línea.

```
<?php
  session_start();
  if (isset($_SESSION['admin'])){
     unset($_SESSION['admin']);
  }
  header('location:index.php');
?>
```

El script permite destruir la variable **admin** registrada en la sesión, se verifica que la sesión se encuentra llena de valores y es destruida con la **function unset**, y direccionada nuevamente a la página origen llamada **index.php**. Este código permitirá destruir la variable de sesión cuando el usuario se equivoque en el ingreso de usuario y clave, o cuando necesite volver a cerrar la sesión.

```
<?php
  session_start();
  require 'encabezado.php';
  echo "<p id='centrado'>";
  echo 'Bienvenido > '.$_SESSION['admin'].'<br>';
  echo "|<a href='cerrar.php'> Cerrar Sesion </a>|";
  echo '</p>';
?>
```

En el siguiente script usamos código puro de PHP para componer la página de bienvenida. Para lograr dicho proceso, enviamos las etiquetas HTML por medio de la instrucción **echo** del PHP, esta interpreta las etiquetas y las muestra de manera adecuada. Debemos tener cuidado con el uso de las comillas dobles y simples, ya que al combinar PHP con HTML ambos usan doble comilla y su uso puede resultar confuso.

Sólo fines educativos - FreeLibros

- **Caso desarrollado 5: Votación de candidatas**

Implemente una aplicación web con PHP que permita controlar las votaciones online de 4 candidatas a un reinado de belleza.

Inicialmente las candidatas deben aparecer según el siguiente formato:

Tener en cuenta:

- Implementar toda la aplicación en una carpeta llamada **Candidatas**.
- Cada candidata debe presentar su nombre completo y su edad.
- Cada candidata tendrá un botón **Votar** para que el usuario pueda seleccionar a la candidata de su preferencia, en el mismo instante se deberá emitir los resultados como «Total de votos», y el porcentaje que este representa frente a las demás candidatas.
- Veamos la interfaz después de realizar las votaciones de algunos usuarios:

Sólo fines educativos - FreeLibros

- Al final deberá mostrar el resumen de las votaciones, el total de votantes y la ganadora hasta el momento junto con el total de votos recaudados.

index.php

```
<!DOCTYPE html>
<html>
    <head>
        <meta charset="UTF-8">
        <title></title>
        <link href="estilo.css" rel="stylesheet">
    </head>
    <body>
        <header>
            <h3 id="centrado">VOTACION DE CANDIDATAS</h3>
            <h2 id="titulo">REYNA DE PRIMAVERA 2015</h2>

        </header>
        <section>
            <?php
                error_reporting(0);
                session_start();

                $total = $_SESSION['total'];
                $pcandidata1 = ($_SESSION['candidata1']*100)/$total;
                $pcandidata2 = ($_SESSION['candidata2']*100)/$total;
                $pcandidata3 = ($_SESSION['candidata3']*100)/$total;
```

Sólo fines educativos - FreeLibros

```
            $pcandidata4 = ($_SESSION['candidata4']*100)/$total;
        ?>
        <form name="frmVotacion" method="POST" action="conteo.php">
            <table border="1" width="600"
                   cellspacing="10" cellpadding="1">
            <tr>
                <td id="centrado"><img src="candidata1.JPG"/></td>
                <td id="centrado"><img src="candidata2.JPG"/></td>
            </tr>
            <tr>
              <td id="centrado">Marisol Romero 19 años <br>
                <input type="submit" value="Votar" name="btnBoton1"/><br>
                TOTAL DE VOTOS: <?php echo $_SESSION['candidata1']; ?><br>
                PORCENTAJE DE VOTOS: <?php echo round($pcandidata1,2);?>%
              </td>
              <td id="centrado">Cesia Alfaro 22 años<br>
                <input type="submit" value="Votar" name="btnBoton2" /><br>
                TOTAL DE VOTOS:<?php echo $_SESSION['candidata2']; ?><br>
                PORCENTAJE DE VOTOS:<?php echo round($pcandidata2,2); ?>%
              </td>
            </tr>
            <tr>
              <td id="centrado"><img src="candidata3.JPG"/></td>
              <td id="centrado"><img src="candidata4.JPG"/></td>
            </tr>
            <tr>
              <td id="centrado">Micaela Borja 20 años<br>
              <input type="submit" value="Votar" name="btnBoton3" /><br>
              TOTAL DE VOTOS:<?php echo $_SESSION['candidata3'];?><br>
              PORCENTAJE DE VOTOS:<?php echo round($pcandidata3,2);?>%
             </td>
             <td id="centrado">Karla Guerrero 21 años<br>
               <input type="submit" value="Votar" name="btnBoton4" /><br>
               TOTAL DE VOTOS: <?php echo $_SESSION['candidata4'];?><br>
               PORCENTAJE DE VOTOS:<?php echo round($pcandidata4,2);?>%
             </td>
            </tr>
            </table>
        </form>
        <?php
            $arreglo = array('Marisol Romero'=>$_SESSION['candidata1'],
                             'Cesia Alfaro'=>$_SESSION['candidata2'],
                             'Micaela Borja'=>$_SESSION['candidata3'],
                             'Karla Guerrero'=>$_SESSION['candidata4']);
            arsort($arreglo);
            reset($arreglo);
            list($candidata,$puntaje)=each($arreglo);
        ?>
        <table border="1" width="600" cellspacing="10" cellpadding="1">
            <tr>
                <td id="ganadora">TOTAL DE VOTANTES:
                                 <?php echo $_SESSION['total']; ?>
                </td>
            </tr>
            <tr>
                <td id="ganadora">GANADORA:<?php echo $candidata; ?>
                    (<?php echo $puntaje; ?> votos)
                </td>
            </tr>
        </table>
    </section>
```

Sólo fines educativos - FreeLibros

```
        <footer>
            <h5 id="centrado">Todos los derechos reservados @2015
                        Diseñado por M@nuel Torres</h5>
        </footer>
    </body>
</html>
```

conteo.php

```
<?php
    session_start();
    if ($_POST['btnBoton1']) $_SESSION['candidata1']+=1;
    if ($_POST['btnBoton2']) $_SESSION['candidata2']+=1;
    if ($_POST['btnBoton3']) $_SESSION['candidata3']+=1;
    if ($_POST['btnBoton4']) $_SESSION['candidata4']+=1;
    $_SESSION['total']=$_SESSION['candidata1']+$_SESSION['candidata2']+
                       $_SESSION['candidata3']+$_SESSION['candidata4'];
    header('location:index.php');
?>
```

estilo.css

```
body {
  background-image:url(fondo.jpg);
  font-family: tahoma;
  font-size: 12px;
}
#centrado{
  text-align: center;
}
table {
    margin:auto;
}
td {
  border: solid 1px #006699;
  border-top: #006699;
  border-right: #006699;
  border-left: #006699;
  border-bottom: #006699;
}
#titulo{
  font-family: tahoma;
  text-align: center;
}
#ganadora{
    font-family: tahoma;
    font-size: 20px;
}
```

Sólo fines educativos - FreeLibros

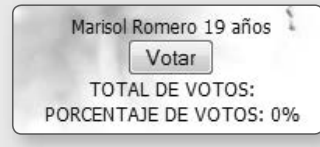
Comentarios:

```
error_reporting(0);
session_start();

$total = $_SESSION['total'];
$pcandidata1 = ($_SESSION['candidata1']*100)/$total;
$pcandidata2 = ($_SESSION['candidata2']*100)/$total;
$pcandidata3 = ($_SESSION['candidata3']*100)/$total;
$pcandidata4 = ($_SESSION['candidata4']*100)/$total;
```

Omitimos los errores de advertencia de PHP e iniciamos la sesión en el archivo **index.php**; luego obtenemos, desde la sesión, los valores registrados en las variables **$total**, **$pcandidata1**, **$pcandidata2**, **$pcandidata3** y **$pcandidata4**.

Consideremos que la variable **$pcandidata1** calcula el porcentaje de cada participante en las votaciones, por tanto la fórmula **($_SESSION['candidata1']*100)/$total** representa el total de votos obtenidos por la primera candidata (**$_SESSION['candidata1']**) multiplicado por 100 y dividido por el número total de votaciones. Esto nos dará el porcentaje de cada candidata.

```
<td id="centrado">Marisol Romero 19 años <br>
    <input type="submit" value="Votar" name="btnBoton1"/><br>
    TOTAL DE VOTOS: <?php echo $_SESSION['candidata1']; ?><br>
    PORCENTAJE DE VOTOS: <?php echo round($pcandidata1,2);?>%
</td>
```



El nombre de la candidata se imprime directamente sobre la celda correspondiente, el botón **Votar** de la misma forma. Recuerde que cada candidata tendrá su propio botón **Votar** de tipo **submit**, así el nombre de dicho botón resultará importante al momento de comparar las votaciones. Finalmente, se imprime el total de votos obtenidos y el porcentaje por cada candidata.

```
$arreglo = array('Marisol Romero'=>$_SESSION['candidata1'],
                 'Cesia Alfaro'=>$_SESSION['candidata2'],
                 'Micaela Borja'=>$_SESSION['candidata3'],
                 'Karla Guerrero'=>$_SESSION['candidata4']);
arsort($arreglo);
reset($arreglo);
list($candidata,$puntaje)=each($arreglo);
```

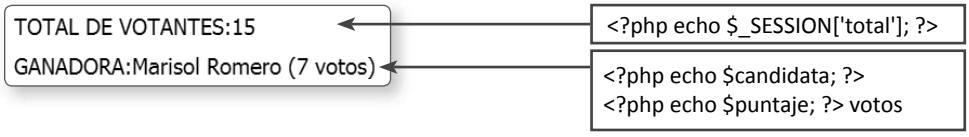
Definimos la variable **$arreglo** que contendrá el número de votos de cada candidata asociada al nombre de la misma. Debemos tener en cuenta que, en la sesión la variable, **candidata1** registra el total de votos de dicha candidata.

Para poder determinar a la ganadora ordenamos el arreglo de manera descendente con la función **arsort**, ubicamos el puntero en el inicio de los registros con la función **reset** y capturamos los datos de la candidata con su respectivo puntaje **list($candidata,$puntaje)=each($arreglo)**.

Sólo fines educativos - FreeLibros

```
<tr>
     <td id="ganadora">TOTAL DE VOTANTES:
         <?php echo $_SESSION['total']; ?>
     </td>
</tr>
<tr>
     <td id="ganadora">GANADORA:<?php echo $candidata; ?>
         (<?php echo $puntaje; ?> votos)
     </td>
</tr>
```

El script presenta la impresión del resultado final de la votación:




```
session_start();
if ($_POST['btnBoton1']) $_SESSION['candidata1']+=1;
if ($_POST['btnBoton2']) $_SESSION['candidata2']+=1;
if ($_POST['btnBoton3']) $_SESSION['candidata3']+=1;
if ($_POST['btnBoton4']) $_SESSION['candidata4']+=1;
$_SESSION['total']=$_SESSION['candidata1']+$_SESSION['candidata2']+
                   $_SESSION['candidata3']+$_SESSION['candidata4'];
header('location:index.php');
```

El archivo **conteo.php** tiene la misión de controlar el número de veces que un usuario realiza una votación; este control se realiza mediante la evaluación del botón **Votar** por cada candidata, de tal forma que:

if ($_POST['btnBoton1']): Evalúa que el usuario haya seleccionado el botón **Votar** de la candidata 1.

$_SESSION['candidata1']+=1: Aumenta en uno el número de votos de la candidata 1 y, a la vez, dicho valor se actualiza dentro de la variable de sesión **candidata1**. También debemos calcular el total de votos ejecutados y lo almacenamos en la variable de sesión **total**; el cual suma todos los votos registrados en las variables de sesión **candidata1**, **candidata2**, **candidata3**, **candidata4**.

Finalmente, se direcciona al archivo inicial **index.php**, ya que el trabajo de este archivo consiste solo en enviar la información a la sesión y no mostrar valores.

Sólo fines educativos - FreeLibros

- **Caso desarrollado 6: Carrito de compras básico – Venta de productos**

Implemente una aplicación web con PHP que controle la venta de productos mediante un carrito de compras básico. El usuario deberá seleccionar un producto de la lista de productos para luego registrar la cantidad de productos a comprar. Inicialmente la interfaz de la venta debe mostrarse de la siguiente manera:

Tener en cuenta:

- Implementar toda la aplicación en una carpeta llamada **VentaCarrito**.
- Los productos a vender ya se encuentran precargados en el control cuadro combinado.
- Al seleccionar un producto del control cuadro combinado deberá mostrarse una imagen del mismo, para esto deberá tener una carpeta adicional al proyecto llamada **fotosProductos**, tal como se muestra a continuación:

Como notamos, cada producto tiene en su nombre un código de producto que luego será asociado a la etiqueta <img> del HTML.

Sólo fines educativos - FreeLibros

- Al seleccionar un producto se mostrará la caja de ingreso para la cantidad de productos a comprar.

- Cada vez que seleccionamos un producto deberá mostrar una imagen del producto seleccionado. Desde aquí ya podemos registrar la cantidad de productos a comprar. El botón **Comprar** permitirá acceder al carrito de compras mostrando el código del producto, descripción, precio, cantidad comprada y el subtotal; tal como se muestra en la siguiente imagen:

- El total a pagar resulta de acumular cada subtotal de cada producto comprado por el usuario.
- En la ventana anterior se mostrará el enlace a «Seguir comprando», que permitirá seleccionar un segundo producto para la compra y así llenar el carrito de compras; también se muestra el enlace «Finalizar la compra» que permitirá anular toda la venta y direccionar al usuario a la página inicial del proyecto. La siguiente imagen muestra el resultado de la compra de dos productos:

Sólo fines educativos - FreeLibros

- El objetivo de cada archivo del proyecto es :

index.php: Presenta la pantalla principal del proyecto.
encabezado.php: Muestra el encabezado que se presentará en cada página del proyecto, aquí es donde se imprime el *banner* de la aplicación.
pie.php: Muestra el pie de página que se presentará en cada página del proyecto.
asignaciones.php: Permite asignar un precio al producto y mostrar la descripción de un determinado producto identificado por su código.
seleccionaProductos.php: Permite seleccionar el producto desde el control cuadro combinado.
destruir.php: Permite descargar las variables de sesión y destruir la sesión para un nuevo registro de venta.
capturaDatos.php: Permite obtener los valores registrados por el usuario, como la descripción y la cantidad a comprar.
canasta.php: Permite añadir un producto al arreglo y mostrar el registro de ventas.
estilo.css: Contiene los estilos que se aplicarán en todos los archivos del proyecto.

Archivo: **index.php**

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="UTF-8">
    <title></title>
    <link href="estilo.css" rel="stylesheet" >
  </head>
  <body>
    <header>
       <?php include 'encabezado.php'; ?>
    </header>
    <section>
```

Sólo fines educativos - FreeLibros

```
        <?php
           error_reporting(0);
           session_start();
           include 'capturaDatos.php';
           $producto = getProducto();

           include 'seleccionaProducto.php';
        ?>
        <form name="frmSeleccion"method="POST" action='index.php'>
          <table border="1" width="550" cellspacing="10" cellpadding="1">
             <tr>
               <td width="200">Seleccione un producto</td>
               <td width="300">
                <select name="selProducto" onchange="this.form.submit()">
                  <option value="p001" <?php echo $selP1;?>>Gaseosa</option>
                  <option value="p002" <?php echo $selP2;?>>
                     Mayonesa en sobre
                  </option>
                  <option value="p003" <?php echo $selP3;?>>
                     Chocolate para niños
                  </option>
                  <option value="p004" <?php echo $selP4;?>>Fideos</option>
                  <option value="p005" <?php echo $selP5;?>>Conservas</option>
                  <option value="p006" <?php echo $selP6;?>>Chocolate</option>
                  <option value="p007" <?php echo $selP7;?>>
                     Cafe 300mg.
                  </option>
                  <option value="p008" <?php echo $selP8;?>>
                     Mayonesa pote
                  </option>
                  <option value="p009" <?php echo $selP9;?>>
                     Crema Dental
                  </option>
                  <option value="p010" <?php echo $selP10;?>>
                     Cubito de pollo
                  </option>
                </select>
               </td>
               <td rowspan="3">
                 <?php
                   if ($_POST[selProducto]) { ?>
                     <img src="fotosProductos/<?php  echo getProducto();?>.jpg"
                          width="120" height="120"/>
               </td>
             </tr>
             <tr>
               <td>Cantidad</td>
               <td><input type="text" name="txtCantidad" value="" /></td>
               <td></td>
             </tr>
             <tr>
               <td><input type="submit" value="Comprar"
                          onclick = "this.form.action = 'canasta.php'"
                          name='btnComprar''/>
               </td>
               <td></td>
               <td></td>
             </tr>
          </table>
        </form>
    </section>
```

Sólo fines educativos - FreeLibros

```
    <footer>
       <?php include('pie.php'); ?>
    </footer>
    <?php } ?>
  </body>
</html>
```

Comentarios:

```
error_reporting(0);
session_start();
include 'capturaDatos.php';
$producto = getProducto();

include 'seleccionaProducto.php';
```

Omitimos los errores de advertencia de PHP, luego iniciamos la session para registrar los productos en el carrito de compras. Incluimos el archivo **capturaDatos.php** que cuenta con dos funciones **getProducto** (captura el producto seleccionado por el usuario) y **getCantidad** (captura la cantidad de productos), luego la variable **$producto** captura el valor devuelto por la función **getProducto**.

Finalmente, incluimos el archivo **seleccionaProducto.php**, que tiene por misión guardar el producto seleccionado desde el control cuadro combinado; es así que asigna la opción SELECTED. Esto servirá para mantener la información visible del producto seleccionado, aun después de presionar el botón **Comprar**.

```
if ($_POST[selProducto]) { ?>
   <img  src="fotosProductos/<?php    echo  getProducto();?>.jpg"  width="120"
height="120"/>
```

Comprobamos que un producto se ha seleccionado para ver su imagen con la instrucción **if($_POST[selProducto])**; hay que tener en cuenta que la implementación del control cuadro combinado tenga el siguiente código:

```
<select name="selProducto" onchange="this.form.submit()">
```

El atributo **onchange** del control cuadro combinado permite activar la respuesta a un evento; en este caso necesitamos que al seleccionar un producto de la lista muestre su imagen respectiva.

Archivo: **capturaDatos.php**

```
<?php
    function getProducto(){
        return $_POST['selProducto'];
    }
    function getCantidad(){
        return $_POST['txtCantidad'];
    }

?>
```

Sólo fines educativos - FreeLibros

Comentarios:

```
function getProducto(){
       return $_POST['selProducto'];
}
function getCantidad(){
       return $_POST['txtCantidad'];
}
```

La función **getProducto** permite obtener el producto seleccionado y **getCantidad** permite obtener la cantidad registrada por el usuario.

Archivo: **seleccionaProducto.php**

```
<?php
    if ($producto=='p001') $selP1='SELECTED'; else $selP1='';
    if ($producto=='p002') $selP2='SELECTED'; else $selP2='';
    if ($producto=='p003') $selP3='SELECTED'; else $selP3='';
    if ($producto=='p004') $selP4='SELECTED'; else $selP4='';
    if ($producto=='p005') $selP5='SELECTED'; else $selP5='';
    if ($producto=='p006') $selP6='SELECTED'; else $selP6='';
    if ($producto=='p007') $selP7='SELECTED'; else $selP7='';
    if ($producto=='p008') $selP8='SELECTED'; else $selP8='';
    if ($producto=='p009') $selP9='SELECTED'; else $selP9='';
    if ($producto=='p010') $selP10='SELECTED'; else $selP10='';
?>
```

Comentarios:

```
if ($producto=='p001') $selP1='SELECTED'; else $selP1='';
```

Evaluamos cada producto para poder asignarle la opción SELECTED, caso contrario dejarlo vacío ya que de todas las opciones presentadas solo uno debe ser el elegido.

Archivo: **asignaciones.php**

```
<?php
    function asignaPrecio($código){
        switch ($codigo){
            case 'p001': return 1.5;
            case 'p002': return 2.5;
            case 'p003': return 11;
            case 'p004': return 1.5;
            case 'p005': return 5;
            case 'p006': return 22;
            case 'p007': return 18.5;
            case 'p008': return 15;
            case 'p009': return 7.5;
            case 'p010': return 1;
        }
    }

    function muestraDescripcion($codigo){
        switch ($codigo){
            case 'p001': return 'Gaseosa';
            case 'p002': return 'Mayonesa en sobre';
            case 'p003': return 'Chocolate para niños';
            case 'p004': return 'Fideos';
            case 'p005': return 'Conservas';
```

Sólo fines educativos - FreeLibros

```
            case 'p006': return 'Chocolate';
            case 'p007': return 'Cafe 300mg.';
            case 'p008': return 'Mayonesa pote';
            case 'p009': return 'Crema Dental';
            case 'p010': return 'Cubito de pollo';

        }
    }
?>
```

Comentarios:

```
function asignaPrecio($código){
```

La función **asignaPrecio** permite asignar el precio a cada producto seleccionado por el usuario; hay que tener en cuenta que para asignar un precio debemos saber el código del producto.

```
function muestraDescripcion($código){
```

La función **muestraDescripcion** permite mostrar el nombre del producto según el código seleccionado por el usuario.

Archivo: **canasta.php**

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="UTF-8">
    <title></title>
    <link href="estilo.css" rel="stylesheet">
  </head>
  <body>
    <header>
      <?php include('encabezado.php');?>
    </header>
    <section>
      <table border="1" width="550" cellspacing="10" cellpadding="0">
        <tr>
          <td>
            <img src="carritoCompras.jpg" width="80" height="80" />
          </td>
        </tr>
        <tr>
           <th>Codigo</th>
           <th>Descripcion</th>
           <th>Precio</th>
           <th>Cantidad</th>
           <th>Subtotal</th>
        </tr>
          <?php
            error_reporting(0);
            session_start();

            include('capturaDatos.php');
            include('asignaciones.php');
            $código =  getProducto();
            $cantidad = getCantidad();
```

Sólo fines educativos - FreeLibros

```
            if(!isset( $_SESSION[$productos])){
               $_SESSION[$productos][$código]=$cantidad;}
               else{
               foreach( $_SESSION[$productos] as $pro => $cant){
                 if ($codigo==$pro){
                    $_SESSION[$productos][$pro]+=$cantidad;
                    $bandera=1;
                 }
                 $total+=$cant;
               }
            }

            if (!$bandera)
               $_SESSION[$productos][$codigo]=$cantidad;

            if (isset( $_SESSION[$productos])){
               $tSubtotal=0;
               foreach ( $_SESSION[$productos] as $cod => $cant) {
                  $subtotal = $cant*asignaPrecio($cod);
                  $tSubtotal+=$subtotal;
          ?>
         <tr>
            <td id="centrado"><?php echo $cod; ?></td>
            <td><?php echo muestraDescripcion($cod); ?></td>
            <td id="derecha">
                <?php echo '$'.number_format(asignaPrecio($cod),2); ?>
            </td>
            <td id="centrado"><?php  echo $cant; ?></td>
          <td id="derecha"><?php echo '$'.number_format($subtotal,2);?></td>
         </tr>
         <?php
                   }
               }
         ?>
         <tr>
          <td id="resaltado">Total a Pagar</td>
          <td></td>
          <td></td>
          <td></td>
         <td id="totales"><?php echo '$'.number_format($tSubtotal,2);?></td>
         </tr>
         <tr>
          <td colspan="4"><?php echo '<a href="index.php">
                         Seguir comprando..!! </a>';?>
          </td>
          <td colspan="4"><?php echo '<a href="destruir.php">
                         Finalizar la compra</a>';?>
          </td>
         </tr>
       </table>
    </section>
    <footer>
      <?php include('pie.php'); ?>
    </footer>
    </body>
</html>
```

Sólo fines educativos - FreeLibros

Comentarios:

```
error_reporting(0);
session_start();

include('capturaDatos.php');
include('asignaciones.php');
$código =  getProducto();
$cantidad = getCantidad();
```

Iniciamos la sesión en el archivo **canasta.php** e incluimos el archivo **capturaDatos.php**, que permite tener acceso a las funciones **getProducto** y **getCantidad**. También se incluye el archivo **asignaciones.php**, que permite asignar un precio al producto seleccionado y mostrar la descripción del producto seleccionado según el código del producto.

Gracias a la inclusión del archivo **capturaDatos.php** podemos obtener el código y la cantidad de los productos seleccionados, y guardarlos en sus variables respectivas.

```
if(!isset( $_SESSION[$productos])){
          $_SESSION[$productos][$código]=$cantidad;}
          else{
          foreach( $_SESSION[$productos] as $pro => $cant){
              if ($codigo==$pro){
                  $_SESSION[$productos][$pro]+=$cantidad;
                  $bandera=1;
              }
              $total+=$cant;
          }
}
```

Condicionamos la existencia del arreglo en la sesión, si no existe se crea con el siguiente código:

```
if(!isset( $_SESSION[$productos])){
          $_SESSION[$productos][$código]=$cantidad;}
```

Caso contrario, recorremos por el arreglo en la sesión y comparamos si se parece al producto que se encuentra por registrar; si hay coincidencias, entonces actualizamos la cantidad acumulándola con la ya existente:

```
foreach( $_SESSION[$productos] as $pro => $cant){
if ($código==$pro){
          $_SESSION[$productos][$pro]+=$cantidad;
          $bandera=1;
      }
      $total+=$cant;
}
```

```
if (!$bandera) $_SESSION[$productos][$código]=$cantidad;

if (isset( $_SESSION[$productos])){
$tSubtotal=0;
foreach ( $_SESSION[$productos] as $cod => $cant) {
    $subtotal = $cant*asignaPrecio($cod);
    $tSubtotal+=$subtotal;
```

Sólo fines educativos - FreeLibros

La variable **bandera** se usa para comparar si el valor que viene es nuevo o ya se encontraba registrado; en caso sea nuevo, solo registra la cantidad sin actualizarlo, usando el siguiente código:

```
if (!$bandera) $_SESSION[$productos][$código]=$cantidad;
```

Finalmente, se calcula el subtotal, para lo cual debemos hacer un recorrido por todo el arreglo y obtener la cantidad y el precio de cada producto, tal como se muestra en el siguiente código:

```
if (isset( $_SESSION[$productos])){
$tSubtotal=0;
foreach ( $_SESSION[$productos] as $cod => $cant) {
    $subtotal = $cant*asignaPrecio($cod);
    $tSubtotal+=$subtotal;
```

La variable **$subtotal** calcula el subtotal de cada producto, mientras que **$tSubtotal** acumula los subtotales para que al final obtengamos cuánto debe pagar el cliente por toda la compra.

Archivo: **destruir.php**

```
<?php
    session_start();
    session_unset();
    session_destroy();
    header("location:index.php");
?>
```

Comentarios:

- session_start(): Permite iniciar la sesión en el archivo destruir.php hay que tener en cuenta que todo archivo que manipule una sesión debe iniciar con esta instrucción.
- session_unset(): Descarga todas las variables contenidas en la sesión.
- session_destroy(): Destruye la sesión incluida todas las variables.
- header("location:index.php"): Redirecciona a la página index.php.

Archivo: **encabezado.php**

```
<!DOCTYPE html>
<html>
    <head>
        <meta charset="UTF-8">
        <title></title>
        <link href="estilo.css" rel="stylesheet" >
    </head>
    <body>
        <h2 id="centrado">CARRITO DE COMPRAS BASICO</h2>
            <h4 id="centrado">COMPRAS ONLINE</h4>
        <img src="banner.JPG" width="550" height="130"/>
    </body>
</html>
```

Sólo fines educativos - FreeLibros

Archivo: **pie.php**

```
<!DOCTYPE html>
<html>
    <head>
        <meta charset="UTF-8">
        <title></title>
        <link href="estilo.css" rel="stylesheet">
    </head>
    <body>
        <h6 id="centrado">Todos los derechos reservados@2015
                          Lic. Manuel Torres</h6>
    </body>
</html>
```

Archivo: **estilo.css**

```
body{
    font-family: tahoma;
    font-size: 14px;
}
#centrado{
    text-align: center;
}
table {
    margin: auto;
}
img{
    margin: auto;
    display: block;
}
td {
    border: solid 1px #006699;
    border-top: #006699;
    border-right: #006699;
    border-left: #006699;
    border-bottom: #006699;
}
table th {
background-color: #b9c9fe;
}
#resaltado{
    font-family: 'trebuchet MS' ;
    font-size: 22px ;
}
#derecha{
    text-align: right;
}
#totales{
    font-family: 'trebuchet MS' ;
    font-size: 22px ;
    text-align: right;
}
```

Sólo fines educativos - FreeLibros

- **Caso desarrollado 7: Registro de nuevos productos**

Implemente una aplicación web con PHP que permita registrar productos de una tienda comercial que vende componentes de una computadora. Inicialmente, la interfaz debe mostrarse de la siguiente manera:

Tener en cuenta:

- Implementar toda la aplicación en una carpeta llamada **RegistroProductos**.
- La aplicación debe registrar la descripción, stock y precio unitario de un determinado producto. Tal como se muestra en la siguiente imagen:

index.php

- Se deben implementar dos botones, el primero llamado **Ver listado de productos**, que permitirá visualizar los productos registrados; y el botón **Registrar producto**, que permitirá registrar el producto y todos sus datos, además de mostrar la lista de productos registrados. Tal como se muestra en la siguiente imagen:

Sólo fines educativos - FreeLibros

listado.php

Al listar los productos registrados o consultados deberá mostrar los enlaces «Seguir registrando», que permitirá volver a la página **index.php**, y el enlace «Cerrar sesión» anulará todo el registro dejando vacía la sesión.

- El objetivo de cada archivo del proyecto es :

index.php: Presenta la pantalla principal del proyecto.
encabezado.php: Muestra el encabezado que se presentará en cada página del proyecto, aquí es donde se imprime el *banner* de la aplicación.
pie.php: Muestra el pie de página que se presentará en cada página del proyecto.
listadoProductos.php: Permite mostrar una lista de los productos que se encuentran en la sesión.
destruir.php: Permite descargar las variables de sesión y destruir la sesión para un nuevo registro.
capturaDatos.php: Permite obtener los valores registrados por el usuario, como la descripción, el stock y el precio del producto.
agregarProducto.php: Permite añadir un producto al arreglo que se encuentra en la sesión, este archivo no visualiza los productos registrados ya que existe un archivo **llamado listadoProductos.php**.
estilo.css: Contiene los estilos que se aplicarán en todos los archivos del proyecto.

Archivo: **index.php**

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="UTF-8">
    <title></title>
    <link href="estilo.css" rel="stylesheet">
  </head>
  <body>
    <header>
       <?php include 'encabezado.php'; ?>
    </header>
    <section>
      <?php
```

Sólo fines educativos - FreeLibros

```
            include('capturaDatos.php');
        ?>
        <form name="frmPrincipal" method="POST" action="agregarProducto.php">
           <table border="1" width="600" cellspacing="10" cellpadding="0">
             <tr>
                <td>Descripción del Producto</td>
                <td><input type="text" name="txtDescripcion"
                     value="<?php echo getDescripcion();?>" size="60"/></td>
             </tr>
             <tr>
               <td>Stock</td>
               <td><input type="text" name="txtStock"
                           value="<?php echo getStock();?>" /></td>
             </tr>
             <tr>
               <td>Precio de producto</td>
               <td><input type="text" name="txtPrecio"
                           value="<?php echo getPrecio(); ?>" /></td>
             </tr>
             <tr>
               <td><input type="submit" name="btnListado"
                      onclick = "this.form.action = 'listadoProductos.php'"
                      value="Ver listado de productos" />
               </td>
               <td><input type="submit" name="btnRegistrar"
                      onclick = "this.form.action = 'agregarProducto.php'"
                      value="Registrar producto" /></td>
             </tr>
           </table>
         </form>
    </section>
    <footer>
         <?php include 'pie.php'; ?>
    </footer>
  </body>
</html>
```

Archivo: **encabezado.php**

```
<!DOCTYPE html>
<html>
    <head>
        <meta charset="UTF-8">
        <title></title>
    </head>
    <body>
        <h2 id="centrado">CONTROL DE PRODUCTOS</h2>
        <img src="banner.JPG" width="600" heigth="100"/>
        <h4 id="centrado">Registro del nuevo producto</h4>
    </body>
</html>
```

Sólo fines educativos - FreeLibros

Archivo: **pie.php**

```
<!DOCTYPE html>
<html>
    <head>
        <meta charset="UTF-8">
        <title></title>
    </head>
    <body>
         <h6 id="centrado">Todos los derechos reservados@2015
                          Lic. Manuel Torres</h6>
    </body>
</html>
```

Archivo: **listadoProductos.php**

```
<?php
    error_reporting(0);
    session_start();
?>
<!DOCTYPE html>
<html>
 <head>
    <meta charset="UTF-8">
    <title></title>
    <link href="estilo.css" rel="stylesheet">
 </head>
 <body>
   <header>
      <?php include 'encabezado.php'; ?>
   </header>
   <section>
      <table border="1" width="600" cellspacing="10" cellpadding="1">
        <tr>
          <th>DESCRIPCION DEL PRODUCTO</th>
          <th>STOCK</th>
          <th>PRECIO</th>
        </tr>
        <?php
          if(isset($_SESSION['misProductos'])){
              foreach ($_SESSION['misProductos'] as $producto) {
        ?>
        <tr>
           <td><?php echo $producto['descripción']; ?></td>
           <td><?php echo $producto['stock']; ?></td>
           <td><?php echo '$'.number_format($producto['precio'],2); ?></td>
        </tr>
        <?php
           }
          }else
             echo '<p id="centrado">No hay productos en la canasta..!!!</p>';
        ?>
      </table>
    </section>
    <footer>
      <p id="centrado">
         <a href="index.php">Seguir comprando..!!</a>|
         <a href="destruir.php">Cerrar sesión</a>
      </p>
      <?php include 'pie.php'; ?>
    </footer>
  </body>
</html>
```

Sólo fines educativos - FreeLibros

Archivo: **destruir.php**

```
<?php
    session_start();
    session_unset();
    session_destroy();
    header('location:index.php');
?>
```

Archivo: **capturaDatos.php**

```
<?php
    function getDescripcion(){
        return $_POST['txtDescripcion'];
    }
    function getStock(){
        return $_POST['txtStock'];
    }
    function getPrecio(){
        return $_POST['txtPrecio'];
    }
?>
```

Archivo: **agregarProducto.php**

```
<?php
    session_start();
    error_reporting(0);
?>
<!DOCTYPE html>
<html>
  <head>
    <meta charset="UTF-8">
    <title></title>
    <link href="estilo.css" rel="stylesheet">
  </head>
  <body>
    <header>
       <?php include 'encabezado.php'; ?>
    </header>
    <section>
      <table border="1" width="600" cellspacing="10" cellpadding="1">
         <tr>
            <th>DESCRIPCION DEL PRODUCTO</th>
            <th>STOCK</th>
            <th>PRECIO</th>
         </tr>
         <?php
            if (isset($_POST['txtDescripcion'])) {
               if (isset($_SESSION['misProductos']))
                  $productos = $_SESSION['misProductos'];

               $productos[$_POST['txtDescripcion']] = array(
                       'descripción' => $_POST['txtDescripcion'],
                       'stock' => $_POST['txtStock'],
                       'precio' => $_POST['txtPrecio']);

               $_SESSION['misProductos'] = $productos;
            }
```

Sólo fines educativos - FreeLibros

```
                header('location:listadoProductos.php');
            ?>
        </table>
    </section>
    <footer>
        <p id="centrado">
            <a href="index.php">Seguir comprando..!!</a>
        </p>
        <?php include 'pie.php'; ?>
    </footer>
  </body>
</html>
```

Archivo: **estilo.css**

```
body{
    font-family: tahoma;
    font-size: 14px;
}
#centrado{
    text-align: center;
}
table {
    margin: auto;
}
img{
    margin: auto;
    display: block;
}
td {
    border: solid 1px #006699;
    border-top: #006699;
    border-right: #006699;
    border-left: #006699;
    border-bottom: #006699;
}
table th {
background-color: #b9c9fe;
}
#resaltado{
    font-family: 'trebuchet MS' ;
    font-size: 22px ;
}
#derecha{
    text-align: right;
}
#totales{
    font-family: 'trebuchet MS' ;
    font-size: 22px ;
    text-align: right;
}
```

Sólo fines educativos - FreeLibros

Sólo fines educativos - FreeLibros

Bibliografía

Cabezas, Luis. (2004). *Manual imprescindible de PHP 5*. España: Anaya Multimedia.

Doyle, Matt. (2010). *Beginning PHP 5.3.* Indianápolis: Wiley Publishing, Inc.

Olsson, Mikael. (2010). *PHP Quick Scripting Reference.* United States of America: Apress.

Vaswani, Vikram. (2007). *PHP Programming Solutions.* United States of America: McGraw-Hill.

Sólo fines educativos - FreeLibros

Impreso en los talleres gráficos de

Surquillo

Sólo fines educativos - FreeLibros